Chapter 20

Public-key cryptography

Introduction

Cryptography may be described as the art of secure real-time communication through an insecure channel, that is, a channel that makes the transmitted message liable to be intercepted by a third party. The way to make the communication secure is to scramble the information through a process known as an **encryption** method. The intended recipient needs to unscramble (**decrypt** or **decipher**) the information on receipt.

Encryption of messages sent by the communicating parties should be fast, that is, real-time. At the same time, encryption should be robust enough that eavesdroppers monitoring the data flow through the channel have no realistic chance to ever decipher the message.

In the context of cryptography, a procedure for implementing a particular method of encryption and decryption is called a **cryptographic system**, or a **cryptosystem**, for short. The history of cryptosystems is long and rich. It ranges from ancient hieroglyphic ciphers (used in the Old Kingdom of Egypt in about 1900 BC) through to simple systems based on alphabet shifting (claimed to have been used by Julius Caesar) and their medieval improvements by Blaise de Vigenère, up to the Enigma code employed by the German military in World War II, to modern cryptosystems to protect communication over the internet.

Before the internet era, cryptosystems relied on a piece of secret information, the **key**, that the sender and the receiver need to share in order to encrypt and decipher the message. For example, the cryptosystem used by Julius Caesar consisted in replacing each letter of the alphabet by the letter that comes three places further down along the alphabet. More generally, any cryptosystem that consists in shifting the alphabet by a given number of places is called a *Caesar cipher*. The key is then the length of the shift, and the cryptosystem assumes that the key is known to the sender for encryption and to the receiver for decryption. The nature of the keys used in Vigenère's system and in the Enigma code was far more complex, and the cryptosystems themselves far more robust than Caesar ciphers, but all of these examples relied on the fact that the key must be known to both the sender and the receiver. This feature is, in fact, a weakness, known as the **key exchange problem**: the key itself is a secret that needs to be exchanged in some form, and it is as liable to being intercepted as the message itself. It took several centuries before anyone questioned the necessity for the sender and the receiver to share the same key. In the mid 1970s, two American mathematicians, Whitfield Diffie and Martin Hellman, began to work on a communication exchange protocol that did not require key exchange: a cryptosystem where the sender uses a key for encryption and the receiver uses another key for decryption. Their intuition that a key exchange could be avoided lies at the basis of what many consider the most powerful breakthrough in the history of cryptography.

In this chapter we will focus precisely on **public-key cryptosystems**. Informally, such systems require two types of key. The two keys, usually in the form of some positive integers, are algebraically linked. One key is used to encrypt a message, often called the **plaintext**, and to produce the corresponding encrypted message, called the **ciphertext**. This key is called a **public key**; anyone who wants to receive a secure communication can generate a public key and make it publicly known, so that anyone who wants to send this recipient a message can encrypt it using this key. The ciphertext is then sent through an agreed channel to the recipient, who decrypts the message using the second key, called a **private key**. The art of secure communication is to devise procedures preventing persons not in possession of the private key (in particular, malicious eavesdroppers monitoring the communication in the channel) to decipher the encrypted message.

It turns out that the procedures used for this purpose are mostly based on certain algebraic techniques that you have encountered earlier in this module: the main ingredients are modular arithmetic and finite fields. Typically, encryption in a public-key cryptosystem is the result of some algebraic operations carried out on the plaintext, suitably coded as a string of numbers. In order to decrypt, one could, in principle, apply the corresponding inverse operations. The encryption procedure, however, is devised in such a way that it is infeasible to find the inverse algebraic operations in real time (in fact, in the lifetime of our universe if one wants an extremely high level of security) without the knowledge of a private key.

Among a variety of public-key cryptosystems that have been proposed in the literature, we will present a sample of four. In Section 1 we introduce two systems based on modular arithmetic, the Rivest–Shamir–Adleman (RSA) system and the Diffie–Hellman system. Section 2 is devoted to cryptosystems based on finite elliptic curves, which can be viewed as abelian groups defined by means of polynomials in two variables over finite fields. Since elliptic curves over more general fields have played an important role in the development of mathematics in the last two centuries, we have included a more detailed introduction to this topic. This chapter concludes with a description of two cryptosystems based on finite elliptic curves, the Diffie–Hellman–ElGamal and the Menezes–Vanstone elliptic cryptosystems.

Chapter guide

This chapter makes considerable use of results obtained earlier in the module as well as in this book. In particular you may find it useful to have Book C readily available for reference. Many of the exercises involve practising numerical techniques, so you may find it helpful to have a simple calculator to hand. For some of the calculations involving elliptic curves, a spreadsheet or other means of evaluating a formula for several inputs would be useful. Note that throughout this chapter GF_p will denote the finite field of p elements, as this avoids confusion with the abelian group $(\mathbb{Z}_p, +)$.

We expect the two sections to require similar amounts of study time.

1 Cryptosystems based on modular arithmetic

This section discusses in some detail two cryptosystems based on modular arithmetic, and, more exactly, on calculations of powers modulo a given positive integer. The two systems we introduce are the RSA and the Diffie–Hellman public-key cryptosystems.

The RSA public-key cryptosystem is named after Ron Rivest, Adi Shamir and Leonard Adleman, who first published the corresponding procedure in 1977. It should be noted that the English mathematician Clifford Cocks discovered an equivalent system in 1973, but this discovery remained classified until 1997.

The RSA system and its variations use a public key that includes a number obtained as a product of two distinct large prime numbers, and its security relies on the fact that the prime factors are kept secret. The algebraic operations that give the encryption use this product, along with other information. The inverse operations needed for decryption turn out to be infeasible in a realistic amount of time unless one knows the two prime factors. Since factorisation of large numbers appears to be an intractable problem, in general RSA systems are regarded as secure.

The second system presented in this section is the Diffie–Hellman public-key cryptosystem. It was first published by Whitfield Diffie and Martin Hellman in 1976. Its story is similar to that of RSA, in that an equivalent system was invented a few years earlier in the UK by Malcolm J. Williamson but was kept classified. The infeasibility of breaking the Diffie–Hellman system relies on a different hard mathematical problem, called the *discrete logarithm problem*.

Subsection 1.1 introduces a few mathematical preliminaries related to modular arithmetic. Subsection 1.2 presents the RSA encryption and decryption schemes (that is, an RSA cryptosystem), followed by two examples in Subsection 1.3. Subsection 1.4 discusses the security of our RSA cryptosystem in some detail and Subsection 1.5 discusses the Diffie–Hellman cryptosystem.

1.1 Algebraic preliminaries

We begin by refreshing your skills in determining the highest common factor of two positive integers. You will recall that this was done in Book A, Chapter 1, Section 5.1 by means of the Euclidean Algorithm. This can be presented as follows.

To determine the highest common factor of two integers a, b such that $2 \leq b < a$, the Euclidean Algorithm proceeds in a series of steps S_i for $i \geq 0$. In order to match our notation with that used in Book A, Chapter 1, we let $r_{-1} = a$ and $r_0 = b$. The input at step S_i is an ordered pair of integers (r_{i-1}, r_i) with $0 < r_i < r_{i-1}$, and the output at S_i is an ordered pair of integers of the form (r_i, r_{i+1}); this output is, at the same time, the input for the next step, S_{i+1}. The body of step S_i consists in performing the Division Algorithm on the pair (r_{i-1}, r_i), that is, in determining the unique positive integer q and the unique integer r_{i+1} such that $0 \leq r_{i+1} < r_i$ and $r_{i-1} = q \times r_i + r_{i+1}$. The output at S_i is the ordered pair consisting of the number we divide by (r_i) and the remainder (r_{i+1}). The process terminates at step S_n if $r_{n+1} = 0$. In summary, the steps are as follows.

The input at step S_0 is the initial pair (a, b), because $a = r_{-1}$ and $b = r_0$.

Step	Input	Division Algorithm	
S_0	(a, b)	$a = q_1 \times b + r_1$	$0 < r_1 < b$
S_1	(b, r_1)	$b = q_2 \times r_1 + r_2$	$0 < r_2 < r_1$
S_2	(r_1, r_2)	$r_1 = q_3 \times r_2 + r_3$	$0 < r_3 < r_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
S_i	(r_{i-1}, r_i)	$r_{i-1} = q_{i+1} \times r_i + r_{i+1}$	$0 < r_{i+1} < r_i$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
S_n	(r_{n-1}, r_n)	$r_{n-1} = q_{n+1} \times r_n + 0$	$r_{n+1} = 0$

We now introduce a variant of this procedure which we will call the **Fast Euclidean Algorithm**. It differs from the Euclidean Algorithm as presented above because it requires pre-checking the input at each step. If the input has the form (r_{i-1}, r_i) and $1 < r_i \leq \frac{r_{i-1}}{2}$, step S_i is as above. However, if $r_i > \frac{r_{i-1}}{2}$ we replace r_i with $r_{i-1} - r_i$. and then run step S_i for this new value of r_i. So the Fast Euclidean Algorithm works exactly like the ordinary Euclidean Algorithm, except that pre-checks on the input are needed at each step, as described.

The validity of the Fast Euclidean Algorithm is proved in the same way as the validity of the standard Euclidean Algorithm, using the argument given in Book A, Chapter 1, Section 1.5 together with the fact that $\mathrm{hcf}(r, s) = \mathrm{hcf}(r, r - s)$ for any two positive integers r, s such that $s < r$. You are asked to prove this fact in the next exercise.

Exercise 1.1

Let r and s be two positive integers such that $s < r$. Prove that

$$\mathrm{hcf}(r, s) = \mathrm{hcf}(r, r - s).$$

The next worked exercise illustrates the Fast Euclidean Algorithm.

Worked Exercise 1.1 *Fast Euclidean Algorithm*

Determine the highest common factor of 257 and 151 with the help of the Fast Euclidean Algorithm.

Solution

Running the standard Euclidean Algorithm to determine $\mathrm{hcf}(257, 151)$ would begin by writing

$$257 = 1 \times 151 + 106.$$

The Fast Euclidean Algorithm, however, requires pre-checking the inputs at each step. Here, since $151 > \frac{257}{2}$, we replace 151 with the difference $257 - 151 = 106$ and take the pair $(257, 106)$ as input for the step S_0. In the computation below this pre-check is done at each step S_i, $i \geq 1$. For easier reading, the entries r_i are underlined once and the remainders r_{i+1} twice.

$$\begin{array}{lrcll} S_0\colon & 257 & = & 2 \times \underline{106} + \underline{\underline{45}} & \text{use } \underline{45} \text{ in } S_1, \text{ as } 45 < \frac{106}{2} \\ S_1\colon & 106 & = & 2 \times \underline{45} + \underline{\underline{16}} & \text{use } \underline{16} \text{ in } S_2, \text{ as } 16 < \frac{45}{2} \\ S_2\colon & 45 & = & 2 \times \underline{16} + \underline{\underline{13}} & \text{use } \underline{3} = 16 - 13 \text{ in } S_3, \text{ as } 13 > \frac{16}{2} \\ S_3\colon & 16 & = & 5 \times \underline{3} + \underline{\underline{1}} & \end{array}$$

Since the remainder in the last step is 1, we may terminate the procedure already at this step and conclude that $\mathrm{hcf}(257, 151) = 1$.

Exercise 1.2

Determine the highest common factor of 91 919 and 19 191 with the help of the Fast Euclidean Algorithm.

As our next revision item, we determine the multiplicative inverse of a given number modulo a given integer. Recall from Theorem 3.4(b) of Book A, Chapter 3, that given two positive integers k and e such that $1 \leq e < k$ and e is coprime to k, there exists a unique positive integer d such that $1 \leq d < k$ and $de \equiv 1 \pmod{k}$. The residue class of d modulo k is precisely the multiplicative inverse of the residue class e modulo k in the ring $(\mathbb{Z}_k, +, \times)$. To simplify the explanation, we will simply say that d is the multiplicative inverse of e modulo k.

In the next worked exercise, we will use the Fast Euclidean Algorithm to determine a multiplicative inverse modulo 257. The method is the one employed in Book A, Chapter 1, Worked Exercise 5.2, where we used the equations in the Euclidean Algorithm in reverse order, except that here we use the equations in the Fast Euclidean Algorithm.

Worked Exercise 1.2 *Multiplicative inverses with the Fast Euclidean Algorithm*

Determine the multiplicative inverse of 151 modulo 257 by reversing the Fast Euclidean Algorithm.

Solution

By the result of Worked Exercise 1.1, we know that the multiplicative inverse to 151 modulo 257 indeed exists. (The alert reader may have realised that this is automatic since 257 is a prime!) We determine the multiplicative inverse by taking the last equation in Worked Exercise 1.1 – the one in step S_3 – and working backwards as in the solution to Worked Exercise 5.2 of Book A, Chapter 1. We will need to take into account the replacements in the Fast Euclidean Algorithm.

We begin by writing the equation in step S_3 in the form

$$1 = 16 - 3 \times 5.$$

We continue by making the 'back substitution' $3 = 16 - 13$ and then we substitute for 13 by isolating the doubly underlined 13 from the equation for S_2. We then rearrange terms so that we end up with 1 expressed as a linear combination of the numbers 13 and 16. We continue to step back through the system of equations until we eventually obtain 1 as a linear combination of 151 and 257. The full working is given below, where 'back substitutions' resulting from a replacement have been underlined for clarity.

$$\begin{aligned}
1 &= 16 - 3 \times 5 = 16 - \underline{(16 - 13)} \times 5 \\
&= 16 \times (-4) + 13 \times 5 \\
&= 16 \times (-4) + (45 - 16 \times 2) \times 5 \\
&= 45 \times 5 + 16 \times (-14) \\
&= 45 \times 5 + (106 - 45 \times 2) \times (-14) \\
&= 106 \times (-14) + 45 \times 33 \\
&= 106 \times (-14) + (257 - 106 \times 2) \times 33 \\
&= 257 \times 33 + 106 \times (-80) = 257 \times 33 + \underline{(257 - 151)} \times (-80) \\
&= 257 \times (-47) + 151 \times 80
\end{aligned}$$

We find that $1 = 257 \times (-47) + 151 \times 80$, which gives

$$151 \times 80 \equiv 1 \pmod{257}.$$

This means that the multiplicative inverse of 151 modulo 257 is 80.

Exercise 1.3

Determine the multiplicative inverse to 19 191 modulo 91 919 by reversing the Fast Euclidean Algorithm.

There is an important lesson to be learned from Worked Exercise 1.2 and Exercise 1.3: if r and s are positive and relatively prime integers such that $1 < r < s$, a multiplicative inverse of r modulo s can be determined quickly. Indeed, when running the Fast Euclidean Algorithm, at each step we made sure that the divisor was never larger than half the dividend. It follows that the number of operations needed to express 1 as a linear combination of r and s using the Fast Euclidean Algorithm takes no more than $\text{int}(\log_2 s)$ divisions, where $\text{int}(x)$ denotes the largest integer that does not exceed x. This is because if $\log_2 s = q$, then $2^q = s$, so we can successively divide s by 2 at most $\text{int}(q)$ times before we obtain 1. Note that in our two worked exercises $\text{int}(\log_2 257) = 8$, but only four divisions were needed.

The power of the Fast Euclidean Algorithm becomes more apparent if you think about multiplicative inverses modulo s when the binary expansion of s requires more than 1000 digits. Such numbers are at least as large as 2^{1000}, which, by the way, is much larger than the estimated number of elementary particles in our universe. Yet determining the multiplicative inverse of any given number relatively prime to s with the Fast Euclidean Algorithm would require no more than 1000 divisions!

We will now prove with an auxiliary result from number theory which is closely related to multiplicative inverses. In Book C, Chapter 9, Subsection 3.1 you encountered Euler's ϕ-function: recall that for an integer n, $\phi(n)$ is the number of positive integers smaller than n and relatively prime to n. In Subsection 4.2 of the same chapter you also learned that if n has a prime factorisation of the form

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_t^{\alpha_t},$$

where p_i $(1 \leq i \leq t)$ are the distinct prime factors of n, then

$$\phi(n) = n(1 - p_1^{-1})(1 - p_2^{-1}) \cdots (1 - p_t^{-1}).$$

In what follows, we will need to work with positive integers n that are products of just two distinct primes, that is, with integers of the form $n = pq$, where p and q are prime numbers and $p \neq q$. Observe that our formula for Euler's ϕ-function applied to $n = pq$ yields

$$\phi(n) = (p - 1)(q - 1).$$

For integers n of this kind we now prove a useful result about multiplicative inverses modulo $\phi(n)$.

Lemma 1.3

Let $n = pq$, where p and q are distinct prime numbers, and let e be an integer such that $1 \leq e < \phi(n)$ and e is relatively prime to $\phi(n)$. If d is the multiplicative inverse of e modulo $\phi(n)$, then

$$x^{de} \equiv x \pmod{n}$$

for all integers x.

Proof Since we are working modulo n, it is sufficient to consider the integers x such that $1 \leq x \leq n-1$. The assumption that d is the multiplicative inverse of e is equivalent to $de \equiv 1 \bmod \phi(n)$, which implies that $de = t\phi(n) + 1$ for some integer t.

If p divides x, then $x^{\phi(n)} \equiv x \,(\text{mod } p)$, since both sides of the congruence are equivalent to $0 \,(\text{mod } p)$. If p does not divide x, Fermat's Little Theorem (Book A, Chapter 4, Theorem 1.1) gives $x^{p-1} \equiv 1 \,(\text{mod } p)$, and so

$$x^{(p-1)(q-1)} \equiv 1 \pmod{p},$$

that is $x^{\phi(n)} \equiv 1 \,(\text{mod } p)$. Similarly, we have that $x^{\phi(n)} \equiv 1 \,(\text{mod } q)$. This implies that both p and q divide $x^{\phi(n)} - 1$. But since p and q are primes, the product $pq = n$ also divides $x^{\phi(n)} - 1$ and so $x^{\phi(n)} \equiv 1 \,(\text{mod } n)$. Taking the tth power of this congruence and then multiplying by x gives

$$x^{t\phi(n)+1} \equiv x \pmod{n},$$

or, equivalently, $x^{de} \equiv x \,(\text{mod } n)$. ■

Exercise 1.4

Prove that $x^{841} \equiv x \,(\text{mod } 899)$ for all integers x.

In the next lemma we prove one more fact about congruences that will be helpful later. This fact is, at the same time, another nice application of Fermat's Little Theorem.

For the proof we need to introduce some terminology that will also be useful later: a non-zero element b of GF_p, the finite field of p elements, is said to be a **square** if there exists an $x \in GF_p$ such that $b = x^2$; such an x is said to be a **square root** of b modulo p. Equivalently, if ξ is any primitive element of GF_p, that is, a generator of the multiplicative group of GF_p, then b is a square if, and only if, b is an even power of ξ. In Definition 1.2 of Book C, Chapter 10, we introduced the term **quadratic residue of** p for a square of GF_p. Remember from Chapter 10, Subsection 1.1 that every non-zero square in GF_p has exactly two square roots, which differ just by sign.

Lemma 1.4

Suppose that p is an odd prime and that $p - 1 = 2^r s$, where s is odd and $r \geq 1$. Then, for every b such that $1 < b < p$, either $b^s \equiv 1 \,(\text{mod } p)$, or there exists a t, $0 \leq t \leq r-1$, such that $b^{2^t s} \equiv -1 \,(\text{mod } p)$.

Proof By Fermat's Little Theorem, for every integer b such that $1 < b < p$ we have $b^{p-1} \equiv 1 \,(\text{mod } p)$, or, equivalently, $b^{2^r s} \equiv 1 \,(\text{mod } p)$. We will now take a square root of this congruence. The square root of 1 mod p can only be one of $+1$ or -1, so that taking the square root gives

$b^{2^{r-1}s} \equiv \pm 1 \pmod{p}$. If the right-hand side of this congruence is -1, there is nothing more we can do. If, however, the right-hand side is $+1$ and $r \geq 2$, we may continue taking the square root again. Repeating this procedure we either reach some t, $0 \leq t \leq r-1$, such that $b^{2^t s} \equiv -1 \pmod{p}$, or we never reach such a t, in which case $b^s \equiv 1 \pmod{p}$. ■

In Subsection 1.4 we will need the contrapositive to this lemma, which we state as a separate corollary because it provides a useful sufficient condition for non-primality of a given integer.

Corollary 1.5

Suppose that k is an odd positive integer, $k \geq 3$, and that $k - 1 = 2^r s$, where s is odd and $r \geq 1$. If there exists an integer b, $1 < b < k$, such that $b^s \not\equiv 1 \pmod{k}$ and $b^{2^t s} \not\equiv -1 \pmod{k}$ for all t, $0 \leq t \leq r-1$, then k is not a prime number.

An integer b which satisfies the conditions of Corollary 1.5 is said to be a **witness of non-primality** for k.

Worked Exercise 1.6 *Witness of non-primality*

It is known that, for any positive integers ℓ and m, $2^\ell + 1$ divides $2^m + 1$ if, and only if, m is an odd multiple of ℓ. With the help of this fact, prove that $b = 2$ is a witness of non-primality for $k = 2^{2^n(2d+1)} + 1$ for any integers $n \geq 0$ and $d \geq 1$.

Solution

In the notation of Corollary 1.5 we have

$$k - 1 = 2^{2^n(2d+1)} \times 1,$$

so that $r = 2^n(2d+1)$ and $s = 1$. For $b = 2$ we have $b^s = b^1 = 2 \not\equiv 1 \mod k$. Let us now test the expression $b^{2^t s} = 2^{2^t}$ for $0 \leq t \leq r-1$. If $2^{2^t} \equiv -1 \mod k$ for some t such that $0 \leq t \leq r-1$, then $k = 2^r + 1$ would be a divisor of $2^{2^t} + 1$. By the fact stated in the formulation of this exercise, this would be possible only if 2^t was an odd multiple of $r = 2^n(2d+1)$, which is not the case as $d \geq 1$ implies that $r \geq 3$, so 2^t must be even. By Corollary 1.5, the integer $b = 2$ is a witness of non-primality of $k = 2^{2^n(2d+1)} + 1$.

The conclusion of Worked Exercise 1.6 does not apply to numbers of the form $F_n = 2^{2^n} + 1$, that is, without the odd factor in the exponent. Recall from Definition 2.7 of Book C, Chapter 9, that the integers F_n are known as the Fermat numbers, and for $n = 0, 1, 2, 3, 4$ they are actually prime numbers, called the Fermat primes.

Exercise 1.5

Find a witness of non-primality for $k = 289$.

We conclude with two results on finite fields. In Subsection 1.5 we will need a fast way of constructing a primitive element of a field GF_p for a prime p. We know from Corollary 3.5(b) of Chapter 18 that the number of primitive elements in the finite field of p elements, GF_p, is $\phi(p-1)$. Our current task is to identify a primitive element. This can be done with the help of the following criterion.

Lemma 1.7

Let p be a prime. A non-zero element $\xi \in GF_p$ is a primitive element of GF_p if, and only if, $\xi^{(p-1)/q} \not\equiv 1 \pmod{p}$ for all prime divisors q of $p-1$.

Proof If ξ is a primitive element, then the smallest positive power j such that $\xi^j \equiv 1 \pmod{p}$ is $p-1$. Since all the exponents $\frac{p-1}{q}$ for prime divisors q of $p-1$ are smaller than $p-1$, we have $\xi^{(p-1)/q} \not\equiv 1 \pmod{p}$ for any prime q dividing $p-1$.

Conversely, if ξ is not a primitive element of GF_p, then $\xi^d = 1$ for some positive integer d which is a proper divisor of $p-1$. Let q be a prime divisor of $\frac{p-1}{d}$. Then $\frac{p-1}{d} = bq$ for some positive integer b, which implies that $\frac{p-1}{q} = db$. But now $\xi^{(p-1)/q} = (\xi^d)^b = 1^b = 1$ in GF_p, and hence $\xi^{(p-1)/q} \equiv 1 \pmod{p}$ for some prime divisor q of $p-1$. ■

We illustrate this result in the next worked exercise.

Worked Exercise 1.8 *Finding a primitive element of GF_p*

Find a primitive element of GF_{193}.

Solution

For the prime $p = 193$ we have $p - 1 = 192 = 2^6 \times 3$, and so the prime divisors of $p-1$ are $q = 2, 3$. By Lemma 1.7, the exponents $\frac{p-1}{q}$ to be considered are 64 and 96.

Although the exponents appear to be high, we will show that the associated computations can be done quickly by the **squaring method**. The trick in this method is to write each exponent as a sum of powers of 2 (this can always be done: simply compute the binary expansion of the exponent). This is simple in our case, as $64 = 2^6$ and $96 = 2^6 + 2^5$. We then compute the powers with exponents of the form 2^j by repeated squaring and assemble the final result by multiplication. Here is how this procedure works in our example.

Since we have no clue as to which elements of GF_{193} might be primitive, we begin by experimenting with small numbers. The first candidate to try is 2. To check whether 2 is a primitive element of GF_{193} with the help of Lemma 1.7, we need to compute the residue classes of $2^{64} = 2^{2^6}$ and $2^{96} = 2^{2^6+2^5} = 2^{2^6} \times 2^{2^5}$ modulo 193. The computation of residue classes modulo 193 can be done even on most basic calculators by partial division. Following the outlined strategy of successive squaring, we obtain the following congruences modulo 193.

$$\begin{aligned}
2^{2^1} &= 2^2 = 4 \equiv 4 \\
2^{2^2} &= 4^2 = 16 \equiv 16 \\
2^{2^3} &= 16^2 = 256 \equiv 63 \\
2^{2^4} &= 63^2 = 3969 \equiv 109 \\
2^{2^5} &= 109^2 = 11\,881 \equiv 108 \\
2^{2^6} &= 108^2 = 11\,664 \equiv 84
\end{aligned}$$

We see that $2^{64} = 2^{2^6} \not\equiv 1 \pmod{193}$, which is promising. However, we find that
$2^{96} = 2^{2^6} \times 2^{2^5} \equiv 108 \times 84 = 9072 = 193 \times 47 + 1 \equiv 1 \pmod{193}$.
Thus, by Lemma 1.7, the element 2 is not primitive in GF_{193}.

The next obvious choice to try is the element 3. But now we find that $3^8 = 81^2 = 6561 \equiv -1 \pmod{193}$, and so $3^{64} \equiv 1 \pmod{193}$.
Lemma 1.7 now immediately implies that 3 is not a primitive element of GF_{193}, either.

The next smallest number to try is 4. But since we have found that 2 is not a primitive element of GF_{193}, neither is 4. (Why?) Thus, let us try our luck with 5. The method of squaring gives the following congruences modulo 193.

$$\begin{aligned}
5^{2^1} &= 5^2 = 25 \equiv 25 \\
5^{2^2} &= 25^2 = 625 \equiv 46 \\
5^{2^3} &= 46^2 = 2116 \equiv -7 \\
5^{2^4} &= (-7)^2 = 49 \equiv 49 \\
5^{2^5} &= 49^2 = 2401 \equiv 85 \\
5^{2^6} &= 85^2 = 7225 \equiv 84 \\
5^{2^6} \times 5^{2^5} &= 84 \times 85 = 7140 \equiv -1
\end{aligned}$$

We see that both 5^{64} and 5^{96} are not congruent to 1 modulo 193, and so by Lemma 1.7 we conclude that 5 is a primitive element of GF_{193}.

An algorithm to compute residue classes by the squaring method can be run on a rather basic calculator: all that is needed are built-in operations of addition, subtraction, multiplication, division, and storage in a memory cell. A formal version of such an algorithm will be given in Worked Exercise 1.12 of Subsection 1.3.

If you know some programming, you might like to try writing down an algorithm at this stage.

Exercise 1.6

Find a primitive element of GF_{257}.

Another preliminary result that we will need towards the end of this chapter, in Subsection 2.5, is Euler's Criterion (Book C, Chapter 10, Theorem 1.5). Euler's Criterion tells us how to recognise squares in the field GF_p, where p is an odd prime. We restate it here for reference, using the terminology and notation introduced in this chapter.

Lemma 1.9 *Euler's Criterion*

Let p be an odd prime and let b be a non-zero element of GF_p. Then b is a square in GF_p if, and only if, $b^{(p-1)/2} \equiv 1 \pmod{p}$, and b is a non-square in GF_p if, and only if, $b^{(p-1)/2} \equiv -1 \pmod{p}$.

Moreover, if $p \equiv -1 \pmod{4}$ and if b is a square in GF_p, then the two solutions of the equation $x^2 = b$ are $x = \pm b^{(p+1)/4}$.

We remark that there is no analogous formula for $p \equiv 1 \pmod{4}$. An efficient way of solving the equation $x^2 = b$ in such a case is the Tonelli–Shanks Algorithm, which is beyond the scope of this chapter.

Worked Exercise 1.10 *Squares and square roots in GF_p*

Determine whether 29 is a square in GF_{199}. If it is, find the solutions of the equation $x^2 = 29$ in GF_{199}.

Solution

For $p = 199$ we have $\frac{p-1}{2} = 99 = 2^6 + 2^5 + 2^1 + 2^0$, and we compute the residue class of $29^{99} \pmod{199}$ by the squaring method as in Example 1.8, but with all congruences modulo 199.

$$\begin{array}{rcrcrcr}
29^{2^1} & = & 29^2 & = & 841 & \equiv & 45 \\
29^{2^2} & = & 45^2 & = & 2025 & \equiv & 35 \\
29^{2^3} & = & 35^2 & = & 1225 & \equiv & 31 \\
29^{2^4} & = & 31^2 & = & 961 & \equiv & -34 \\
29^{2^5} & = & (-34)^2 & = & 1156 & \equiv & -38 \\
29^{2^6} & = & (-38)^2 & = & 1444 & \equiv & 51
\end{array}$$

This implies $29^{99} = 51 \times (-38) \times 45 \times 29 \equiv 1 \pmod{199}$, so 29 is indeed a square in GF_{199}. Since $199 \equiv -1 \pmod{4}$, we can actually determine the two solutions of $x^2 = 29$ in GF_{199} by applying the second part of Lemma 1.9 with the exponent $\frac{p+1}{4} = 50 = 2^5 + 2^4 + 2^1$ and using the above table to obtain

$$x = \pm 29^{2^5+2^4+2^1} \equiv (-38) \times (-34) \times 45 \equiv \pm 32.$$

Note that computations related to Euler's Criterion may be simplified further using Legendre symbols, as you saw in Book C, Chapter 10, Subsection 2.2.

Exercise 1.7

Determine whether -1, 3 and -5 are squares in GF_{193}. If they are, find their square roots.

We conclude this subsection with an important observation: when we use the squaring method to check whether a given element $k \in GF_p$ is a primitive element (or a square), the number of lines in the tabular form of the calculations is never larger than $\text{int}(\log_2 k)$. This means that even for enormously large values of k such as, say, $k \approx 2^{1000}$, we would obtain an answer in approximately a 1000-line table, that is, using a constant times 1000 arithmetic operations. Since such operations can be carried out extremely quickly on modern computers, checking for primitive elements or squares can be done in less than a blink of an eye even for numbers bigger than the estimated number of elementary particles in our universe! The same applies to multiplicative inverses of elements, as we noted in the short discussion after Exercise 1.3.

1.2 The RSA cryptosystem

Encryption and decryption procedures are usually described with the help of three characters: Alice (A), who needs to send a message to Bob (B) through an insecure channel where the data flow can be freely monitored by Eve (E), an eavesdropper.

Modern cryptosystems are designed for computer communication, and a message first needs to be converted into a form suitable for transfer through a digital channel. Most channels are binary, which essentially means that the signal flow through the channel can be imagined as a flow of 0s and 1s. Thus, messages need to be converted into binary numbers. However, to simplify matters and avoid doing binary arithmetic we will assume that messages are converted into decimal numbers.

There are numerous ways in which such a conversion can be done. One is to use some standard system, such as the ASCII code for representing capital and small letters of the alphabet, numerals, commas, etc. Another way could consist, for instance, in Alice and Bob agreeing to represent the letters A to Z by the numbers 1 to 26, and to represent a space by the number 27, say. So, for example, the message

'I NEED YOU'

would be represented by the number

09 271 405 050 427 251 521.

When the message to be transmitted is lengthy, it may be necessary to divide the resulting number into smaller pieces, but we will not go into such detail here. The point at issue is that the cryptosystems that we will look at in fact require numerical techniques.

The science of converting plaintext messages into streams of ciphers (typically, binary digits) is *coding theory*, which is a vast and exciting area of mathematics. It differs from cryptography in that the main concern of coding theory is the development of methods for detection and correction of errors in the data flow, rather than the protection of data from eavesdropping.

With the algebraic preliminaries of Subsection 1.1, we are ready to describe the RSA public-key cryptosystem.

Communication setup

In order to start the procedure for secure communication, Bob needs to set up a pair of parameters. He chooses two distinct odd primes p and q and calculates their product $n = pq$. Since he knows the prime factorisation of n, he determines the value of Euler's ϕ-function $\phi(n) = (p-1)(q-1)$. Bob then chooses an integer e that is relatively prime to $\phi(n)$ and such that $1 < e < \phi(n)$. He then releases the pair (n, e) through the communication channel so that everybody, including Alice and Eve, can get hold of this information.

We emphasise that Bob does not release the two primes p, q. In fact, he can discard them, since for communication he will only need the public key (n, e) and the value of $\phi(n)$. The final preparatory move for Bob is to determine the multiplicative inverse to e modulo $\phi(n)$, that is, the unique d such that $1 < d < \phi(n)$ and $de \equiv 1 \pmod{\phi(n)}$. This can be done with the help of the Fast Euclidean Algorithm as in Worked Exercise 1.2. Unlike for the pair (n, e), Bob does not release the value of d.

We sum up the list of required parameters as a definition.

Definition 1.11 *RSA triple, public key, private key*

Let $n = pq$ be a product of two distinct primes p, q and let e be a positive integer such that $1 < e < \phi(n)$ and e is relatively prime to $\phi(n)$. Let d be the unique positive integer such that $1 < d < \phi(n)$ and $de \equiv 1 \pmod{\phi(n)}$. Then:

(a) the triple (p, q, e) is called an **RSA triple**

(b) the pair (n, e) is the **public key**

(c) the integer d is the **private key**.

The reason why the pair (n, e) is called the public key and the integer d is called the private key is that Bob releases the pair (n, e) to the public, but keeps d for himself.

We are now ready to describe the encryption and decryption procedures.

Encryption

The encryption procedure depends only on the public key (n, e). Alice begins by converting her message into a number m such that $1 < m < n - 1$; we have mentioned ways to do this earlier. As the next step, Alice uses the public key (n, e) to compute the number

$$c \equiv m^e \pmod{n}.$$

Finally, she sends c to Bob through an appropriate channel. Note that we have deliberately chosen the notation so that m stands for *m*essage and c for *c*iphertext.

Decryption

From the point of view of abstract algebra, the decryption procedure is astonishingly simple. Bob receives c and, knowing n and d, calculates the value $c^d \pmod{n}$. What does he obtain as a result? The 'insider information' that $c = m^e \pmod{n}$ combined with Lemma 1.3 gives

$$c^d \equiv (m^e)^d = m^{de} \equiv m \pmod{n}$$

and hence $c^d \pmod{n}$ is equal to m. In other words, by raising c to the power equal to his private key, Bob has decrypted Alice's message!

Eavesdropping?

Eve, the eavesdropper, can intercept n, e and c by monitoring the communication through the channel. Assuming that she knows how RSA works, is she in a position to decipher the message m with only the knowledge of n, e and c? The answer is in the affirmative if she can factor n: if she can find p and q, she can determine $\phi(n)$ and hence the inverse d of e modulo $\phi(n)$, and then she can decrypt the ciphertext exactly as Bob can. However, in Subsection 1.4 you will see that, in general, factorising n into his prime factors is not feasible, provided the primes p and q chosen by Bob are sufficiently large. In the same subsection we will also discuss other hypothetical ways to 'break' RSA.

1.3 Examples of RSA cryptosystems

We illustrate RSA encryption and decryption in detail in the next example.

Example 1.12 *RSA encryption and decryption*

Suppose that Alice needs to send Bob the message 619, which is the three-digit security number on her Visa credit card, for him to complete an online purchase. To avoid lengthy calculations in this model example, we assume that Bob chooses two small primes, say $p = 61$ and $q = 97$. The product is then $n = pq = 5917$ and the corresponding value of Euler's ϕ-function is

$$\phi(n) = (p - 1)(q - 1) = 5760.$$

Bob now needs to choose an exponent e that is relatively prime to 5760 and is such that $1 < e < 5760$, and then determine the multiplicative inverse of e modulo 5760. For relatively small numbers, such as the ones in this example, Bob can carry out both tasks simultaneously: to choose the exponent e and have its multiplicative inverse d modulo 5760 at the same time, Bob can successively try to factor the integers of the form $5760t + 1$ for $t = 1, 2, \ldots$. From the factorisations he obtains in this way, he chooses any two e and d such that $de = 5760t + 1$. He finds that $t = 8$ gives

$$5760 \times 8 + 1 = 46\,081 = 7 \times 29 \times 227 = 203 \times 227,$$

which is a factorisation he likes. So he chooses $e = 203$ and releases the public key $(n, e) = (5917, 203)$, keeping the private key $d = 227$ for himself.

Alice's message is her three-digit security number, $m = 619$. Using the public key, she now needs to determine the unique positive integer c such that $c \equiv m^e \pmod{n}$, that is, $c \equiv 619^{203} \pmod{5917}$. This calculation might look scary at a first glance, but we will show that it can be performed quickly even using a very modest calculator.

The key is to use the squaring method: Alice writes the exponent 203 as a sum of powers of 2, that is,

$$203 = 128 + 64 + 8 + 2 + 1 = 2^7 + 2^6 + 2^3 + 2^1 + 2^0.$$

Then, letting $m_0 = m$, she successively determines

$$\begin{aligned}
m_1 &= m_0^2 \equiv m^{2^1} \pmod{n} \\
m_2 &= m_1^2 \equiv m^{2^2} \pmod{n} \\
m_3 &= m_2^2 \equiv m^{2^3} \pmod{n} \\
&\vdots \qquad \vdots \qquad\quad \vdots \\
m_7 &= m_6^2 \equiv m^{2^7} \pmod{n}.
\end{aligned}$$

Finally, she computes

$$c \equiv m^{203} = m^{2^7+2^6+2^3+2^1+2^0} \equiv m_7 m_6 m_3 m_1 m_0 \pmod{n}.$$

Alice can do this quickly with the help of a basic calculator. She stores $n = 5917$ in the memory and keeps to the following simple algorithm.

(0) Set $i = 1$; the value of i is your counter (which can be recorded, say, on a piece of paper).

(1) Input m_{i-1} and compute m_{i-1}^2 (by, say, multiplying m_{i-1} by itself), leaving the result on the display.

(2) Divide m_{i-1}^2 by n (to enter n use the 'Read Memory' key).

(3) From the result on the display, subtract its integer part; this leaves on the display a number greater than 0 and smaller than 1.

(4) Multiply the displayed number by n; if necessary, round the result to the nearest integer.

(5) The number on the display is now m_i; record this result.

(6) If $i = 7$, then stop; otherwise increase i by 1 and go to step (1).

Note that steps (3) and (4) simply determine the remainder after partial division by n. The final step, that is, determination of $c \pmod n$, can again be performed using steps (3) and (4) appropriately.

Alice's encryption

Alice runs the algorithm we have just described with the help of her calculator and records the following partial results modulo 5917:

$$\begin{aligned}
&m_0 = 619,\\
&m_0^2 = 383\,161, && m_1 \equiv m_0^2 \equiv m^{2^1} \equiv 4473\\
&m_1^2 = 20\,007\,729, && m_2 \equiv m_1^2 \equiv m^{2^2} \equiv 2352\\
&m_2^2 = 5\,531\,904, && m_3 \equiv m_2^2 \equiv m^{2^3} \equiv 5426\\
&m_3^2 = 29\,441\,476, && m_4 \equiv m_3^2 \equiv m^{2^4} \equiv 4401\\
&m_4^2 = 19\,368\,801, && m_5 \equiv m_4^2 \equiv m^{2^5} \equiv 2460\\
&m_5^2 = 6\,051\,600, && m_6 \equiv m_5^2 \equiv m^{2^6} \equiv 4426\\
&m_6^2 = 19\,589\,476, && m_7 \equiv m_6^2 \equiv m^{2^7} \equiv 4206.
\end{aligned}$$

Having computed all the ingredients, Alice determines the product $m^{203} \equiv m_7 m_6 m_3 m_1 m_0 \pmod{5917}$ as follows:

$$\begin{aligned}
&m_7 m_6 = 18\,615\,756 \equiv 874 \pmod{5917}\\
&m_7 m_6 m_3 = 874 m_3 = 4\,742\,324 \equiv 2807 \pmod{5917}\\
&m_7 m_6 m_3 m_1 = 2807 m_1 = 12\,555\,711 \equiv 5754 \pmod{5917}.
\end{aligned}$$

The final result is

$$c = m_7 m_6 m_3 m_1 m_0 \equiv 5754 m_0 = 3\,561\,726 \equiv 5609 \pmod{5917}.$$

It follows that the ciphertext that Alice transmits to Bob is $c = 5609$.

Bob's decryption

Bob too uses the squaring method to write his private key $d = 227$ as a sum of powers of 2:

$$227 = 128 + 64 + 32 + 2 + 1 = 2^7 + 2^6 + 2^5 + 2^1 + 2^0.$$

Letting $c_0 = c$, he successively computes $c_1 \equiv c^{2^1} \pmod n$, $c_2 \equiv c_1^2 = c^{2^2} \pmod n$, $c_3 \equiv c_2^2 = c^{2^3} \pmod n$, and so on, up to $c_7 \equiv c_6^2 = c^{2^7} \pmod n$, and using these he finally determines

$$c^{227} = c^{2^7+2^6+2^5+2^1+2^0} \equiv c_7 c_6 c_5 c_1 c_0 \pmod n.$$

Through these individual steps, Bob obtains the following results modulo 5917:

$$\begin{aligned}
&c_0 = 5609,\\
&c_0^2 = 31\,460\,881, && c_1 \equiv c_0^2 \equiv c^{2^1} \equiv 192\\
&c_1^2 = 36\,864, && c_2 \equiv c_1^2 \equiv c^{2^2} \equiv 1362\\
&c_2^2 = 1\,855\,044, && c_3 \equiv c_2^2 \equiv c^{2^3} \equiv 3023\\
&c_3^2 = 9\,138\,529, && c_4 \equiv c_3^2 \equiv c^{2^4} \equiv 2681\\
&c_4^2 = 7\,187\,761, && c_5 \equiv c_4^2 \equiv c^{2^5} \equiv 4523\\
&c_5^2 = 20\,457\,529, && c_6 \equiv c_5^2 \equiv c^{2^6} \equiv 2460\\
&c_6^2 = 6\,051\,600, && c_7 \equiv c_6^2 \equiv c^{2^7} \equiv 4426.
\end{aligned}$$

He can now compute $c^{227} \equiv c_7c_6c_5c_1c_0 \pmod{5917}$:

$$c_7c_6 = 10\,887\,960 \equiv 680 \pmod{5917}$$
$$c_7c_6c_5 \equiv 680c_5 = 3\,075\,640 \equiv 4717 \pmod{5917}$$
$$c_7c_6c_5c_1 \equiv 4717c_1 = 905\,664 \equiv 363 \pmod{5917}.$$

The final result is

$$c^{227} \equiv c_7c_6c_5c_1c_0 \equiv 363c_0 = 2\,036\,067 \equiv 619 \pmod{5917},$$

so $m = 619$, which is indeed Alice's three-digit security number!

It is important to note that both Alice's encryption and Bob's decryption can be carried out quickly. Indeed, in general, if m is Alice's message and $1 < m < n$, then one needs at most $\text{int}(\log_2 m)$ squarings modulo n, and then at most another $\text{int}(\log_2 m)$ multiplications modulo n. Since $m < n$, one needs to perform no more than $2\ \text{int}(\log_2 n)$ multiplications modulo n to encrypt an arbitrary message! So RSA encryption and decryption can be carried out rapidly even for extremely large numbers – in less than a blink of an eye on a rather low-performance notebook.

Exercise 1.8 *RSA encryption and decryption*

Alice wants to transmit the message $m = 1111$ to Bob using the RSA cryptosystem with public key $(5963, 203)$.

(a) Determine Alice's ciphertext c.

(b) Given that Bob's private key $d = 515$, verify that his decryption produces Alice's message.

1.4 How secure are RSA cryptosystems?

Let us now examine the security of RSA cryptosystems in more detail. We mentioned the threat posed by a potential eavesdropper, Eve, at the end of Subsection 1.2. What are Eve's chances of deciphering the message m by eavesdropping? By monitoring communication through the channel, Eve can get hold of the public key (n, e) and of the ciphertext, that is, the integer c. We may safely assume that Eve has passed M303 with a distinction, and so she is fully familiar with RSA encryption. This means that Eve knows that Alice's message, an integer m such that $1 < m < n$, has been raised to the power e. Eve also knows that c, the ciphertext sent through the channel, is the remainder of the partial division of m^e by n, where n is known. The question now is whether this knowledge is enough for Eve to determine m.

Eve could certainly decrypt Alice's message if she knew Bob's private key d, as then she would simply follow the decryption procedure. So our question is really whether Eve can determine d from n and e. Since Eve knows that d is the multiplicative inverse of e modulo $\phi(n)$, it would be sufficient for her to determine $\phi(n)$ and then use the Euclidean Algorithm to calculate d as we did in Worked Exercise 1.2.

At this stage you might think that determining $\phi(n)$ should be easy. Since Eve knows the workings of RSA, she also knows that n is the product of two distinct primes p and q. Thus, it is sufficient for Eve to factor n into p and q, and hence recover $\phi(n) = (p-1)(q-1)$.

And now comes the important fine point. Suppose Bob chooses p and q in such a way that their product, n, is enormously large – say, requiring about 1000 bits (binary digits) to be represented in the binary system. To translate this into the decimal system, the approximation $2^{10} \approx 10^3$ tells us that a 1000-bit number would require about 300 decimal digits to be written in decimal notation. Let us assess the feasibility of factoring such a large number.

Prime factorisation: a naive approach

If n is not a prime, it must contain a prime factor not exceeding $\sqrt{n}$. By way of an example, assume that n is only a 300-bit number (that is, with about 90 decimal digits) and let us look for a 150-bit prime factor of n.

In total, there are $t = 2^{150} - 2^{149} = 2^{149}$ 150-bit numbers. As a consequence of the Prime Number Theorem (see Book A, Chapter 2, Subsection 4.3), the number of 150-bit primes is approximately $\frac{t}{\log t}$. We now make a number of rather crude estimates to simplify the numbers we will be working with. For $t = 2^{149}$ we have

$$\frac{t}{\log t} > \frac{t}{\log_2 t} = \frac{2^{149}}{149} > 2^{140} = (2^{10})^{14} > (10^3)^{14} = 10^{42}.$$

So there would be approximately 10^{42} 150-bit primes not exceeding $\sqrt{n}$.

Let us, rather unrealistically, assume that we were able to generate all such primes, so that we could run a trivial algorithm to factorise n by successively dividing n by each of these primes.

If we used a science-fiction super-computer capable of carrying out a single division in one picosecond, that is, in 10^{-12} seconds, in one second we would be able to divide by 10^{12} (a thousand billion) 150-bit primes. In the worst case, however, we would have to try to divide by all the (at least) 10^{42} such primes. This would take $\frac{10^{42}}{10^{12}} = 10^{30}$ seconds. Realising that one year has about 31.5 million (that is, approximately 3.15×10^7) seconds, you see that even under our optimistic assumptions the computation could take longer than 10^{22} years in the worst case! Even a million such computers would not help much – the computing time would be reduced to (longer than) $\frac{10^{22}}{10^6} = 10^{16}$ years!

Prime factorisation: the general number sieve algorithm

The prime factorisation algorithm that we have just described is rather naive, as to factorise n we simply test for all possible prime factors not exceeding $\sqrt{n}$. Arguably, better factorisation algorithms should be available. Currently, the best algorithm available is the **general number sieve algorithm**. This is a very sophisticated procedure well beyond the scope of this module and here we can only give a general idea of how the algorithm operates.

The idea behind the general number sieve algorithm is as follows. To find a prime factor of a given positive integer n, the algorithm uses the theory of algebraic extensions of the ring of integers to produce, in a highly non-trivial way, a pair of positive integers $x > y$ such that n divides $x^2 - y^2$ and $n \neq x + y, x - y$. If p is a prime factor of n, then p is a divisor of $x - y$ or $x + y$. The Fast Euclidean Algorithm determines $\mathrm{hcf}(n, x - y)$ and $\mathrm{hcf}(n, x + y)$; one of these must contain p as a factor.

If n is a number of at least 100 decimal digits, it has been estimated that, in the worst case, factorising n using the general number sieve algorithm requires about $\exp[c(\log n)^{1/3}(\log(\log n))^{2/3}]$ 'operations', where c is a constant. We have put the word 'operations' in quotation marks because the value of the constant c depends on what one understands by an 'operation'. Let us assume that an 'operation' is either addition, subtraction, multiplication or division. For simplicity we will rather unrealistically assume that each of these operations takes the same amount of time. The theory underlying the algorithm then shows that the constant c may be replaced by 2 for, say, 1000-digit numbers.

Now suppose that the magnitude of n is approximately 2^{1000}, that is, the binary representation of n requires about 1000 digits (and hence the decimal representation requires about 300 digits). Since $\log_2 n \approx 1000$ and $\log 2 \approx 0.7$, a quick calculation shows that

$$\log n \approx 1000 \log 2 \approx 700 \quad \text{and} \quad \log(\log n) \approx 6.55.$$

For $c = 2$ this approximation gives

$$2(\log n)^{1/3}(\log(\log n))^{2/3} \approx 70.$$

By the above estimate we may assume, for the sake of an example, that the general number sieve algorithm needs no more than $e^{70} \approx 2^{100} \approx 10^{30}$ operations to factor n. If we were to run the algorithm on $3000 = 3 \times 10^3$ machines, and if a single operation took 10^{-9} seconds on each of these machines, in the worst case factoring n would take about $10^{30}/(3 \times 10^3 \times 10^9 \times 3.15 \times 10^7) \approx 10^9$ years. This is much shorter than the time required by the naive algorithm, and for a much larger number, but still very long!

The above estimates were made under numerous simplifications and still refer to the worst case, so that the actual running time for a concrete n might be considerably shorter. The greatest success of the general number sieve algorithm, implemented with a number of further mathematical subtleties, has been the factorisation of a 232 decimal digit number on hundreds of machines over a span of more than 2 years, finished in 2009.

The current recommendation for safe RSA communication is to use numbers that are approximately 1000 bits long, and approximately 2000 bits long if super extra security is required.

Generating large primes: The Rabin–Miller test

The next question we need to address is how to generate large prime numbers so as to find suitable parameters for RSA. Our previous discussion suggests that simply taking a random large number (say, 200 digits long) and trying to factor it is infeasible, so we cannot hope to find large primes in this way. Luckily, there appears to be a huge difference between testing a number for primality and factorising it into its prime divisors.

To explain the idea of primality testing without factorisation, let us assume that we want to test a (large) odd positive integer $k \geq 3$. Since k is odd, there are positive integers r and s such that s is odd and $k - 1 = 2^r s$. Now take an integer b such that $1 < b < k$ and consider the following statement $S(b)$:

$$S(b): \; b^s \not\equiv 1 \,(\mathrm{mod}\ p) \text{ and } b^{2^t s} \not\equiv -1 \,(\mathrm{mod}\ p) \text{ for all } t \text{ such that } 0 \leq t \leq r-1.$$

Recall that, in Subsection 1.1, a positive integer b for which $S(b)$ is a true statement was called a **witness of non-primality** for k. The next lemma explains this choice of terminology.

Lemma 1.13 *Witness of non-primality*

Let k be an odd positive integer, $k \geq 3$, and let $k - 1 = 2^r s$ where r and s are positive integers and s is odd. If the statement $S(b)$ is true for some b such that $1 < b < k$, then k is not a prime.

Proof This is just a re-statement of Corollary 1.5 of Subsection 1.1. ■

Now, what happens if the statement $S(b)$ is false for, say, j distinct random integers b such that $1 < b < k$? We cannot be sure that k is a prime in such a case. However, it can be shown that in this case the probability that k is *not* a prime is less than 4^{-j}. This is the basis of the **Rabin–Miller test**; integers k that have passed the test, that is, such that $S(b)$ is false for j values of b, are called **strong probable primes to base** b.

The Rabin–Miller test

To test a given positive integer k for primality, we choose a positive integer j. We then randomly generate j distinct integers b such that $1 < b < k$. If $S(b)$ is false for all the j values of b, we accept k as a strong probable prime, and then the probability that k is a prime is at least $1 - 4^{-j}$.

Thus, for example, if k is a 150-bit number k and $S(b)$ is false for $j = 40$ randomly chosen values of b strictly between 1 and k, then the probability of k failing to be a prime is less than $2^{-80} < 10^{-24}$. For practical purposes we may therefore treat the strong positive prime k as if it was a prime number.

The estimates we have presented are rather crude, yet they are enough to show the power of the mathematical theory behind the Rabin–Miller test. In practice, testing whether a large, random odd, say 150-bit number k is prime requires generating just about 20 numbers b strictly between 1 and k. If $S(b)$ is false for all the generated bs, then we declare k to be a prime, and we will be right with very high probability. If $S(b)$ happens to be true for just one of these bs, then we need to generate another k and repeat the Rabin–Miller test.

The question remains of how quickly we can generate a number that would pass the Rabin–Miller test. We illustrate the answer by way of an example, again. Let us go back to our 150-bit numbers and recall that among the $t = 2^{149}$ such numbers there are approximately $\frac{t}{\log t}$ primes. Another important result on the distribution of primes says that the average gap between two consecutive primes has length approximately equal to $\log t$. Now observe that

$$\log t = \log(2^{149}) = 149 \log 2 \approx 100.$$

So if a random, odd 150-bit number k is not a prime, one of the numbers $k+2, k+4, \ldots, k+100$ will be a prime with very high probability. It follows that, on the average, the Rabin–Miller test will return a 150-bit number that can be considered a prime after at most 50 trials! In practice, about half of this theoretical average number of trials is usually sufficient.

The Rabin–Miller test can be performed rapidly, since powers modulo k can be computed rapidly, as you saw in the solution to Example 1.12.

We would like to emphasise the difference between primality testing and factorisation again, using a slightly different point of view. Let k be an enormously large odd number such that $k - 1 = 2s$, where s is odd. Assume that k fails the Rabin–Miller test. In this particular case, failing the test means that a number b strictly between 1 and k has been generated such that $b^s \equiv \pm 1 \pmod{k}$. Then, by Corollary 1.5, we know that k is not a prime. Nevertheless, we saw that finding the actual factorisation of such a k is infeasible!

Other ways of attacking the RSA cryptosystem?

Recall that we actually want to find out whether Eve can determine $\phi(n)$ from n without knowing the factorisation pq. Could there be some indirect way of revealing the value of $\phi(n)$? If there were one, then, from $n = pq$ and $\phi(n) = pq - (p+q) + 1$ we would obtain two equations in two unknowns p and q, namely

$$p + q = n + 1 - \phi(n) \quad \text{and}$$
$$pq = n,$$

from which we would be able to determine p and q. But since we saw that there is no 'easy' way to determine the two primes, there cannot be an easy way to determine $\phi(n)$.

A deeper analysis, beyond the scope of this module, actually shows that if there was any 'easy' indirect way of deciphering a general ciphertext c just on the basis of the public key (n, e), then there would be an 'easy' way to determine the two prime factors of n. Since the latter is infeasible for extremely large n, the practical conclusion is that RSA is 'reasonably safe in general'. The addendum 'in general' means that one cannot completely exclude instances when RSA is unsafe, but these occur with negligible probability.

1.5 The Diffie–Hellman cryptosystem

In this subsection we will look at another cryptosystem based on modular arithmetic. Its security, however, is not be based on the infeasibility of prime factorisation, but on the so-called *discrete logarithm problem*, which we will introduce in the course of our explanation.

Communication setup

As before, our two characters Alice and Bob want to communicate securely through an insecure channel. They agree on using a prime p and all the necessary calculations for communication will be done in the field GF_p, or, equivalently, modulo p. For a fast generation of large integers that are prime with high probability, they use the Rabin–Miller test from Subsection 1.4. They further agree a fixed primitive element $\xi \in GF_p$. We have seen that a primitive element can be generated quickly with the help of Lemma 1.7 and the process was illustrated in the solution to Worked Exercise 1.8. The pair (p, ξ) is the public key that Alice and Bob release.

As the next step, Alice chooses a positive integer a such that $1 < a < p - 1$, and Bob chooses a positive integer b such that $1 < b < p - 1$. The integer a is Alice's private key and b is Bob's private key.

The communication is initiated as follows. Alice calculates $\alpha = \xi^a$ in GF_p, or, equivalently, $\alpha \equiv \xi^a \pmod{p}$ with $1 \leq \alpha \leq p - 1$, and sends the number α over the channel to Bob. Symmetrically, Bob calculates $\beta = \xi^b \bmod p$ and sends β to Alice. The numbers α and β are sent through an insecure channel, and so can become public knowledge.

Still as part of the setup, Alice computes β^a in GF_p using her private key a and Bob computes α^b with the help of his private key b. Observe that they must obtain the same result, since

$$\beta^a = (\xi^b)^a = (\xi^a)^b = \alpha^b.$$

We will denote the common value $\beta^a = \alpha^b = \xi^{ab}$ by σ. The value of σ will not be released during the communication; it is a **shared secret** between Alice and Bob.

Now, both Alice and Bob can determine σ^{-1}, the multiplicative inverse of σ in GF_p, and they can do it in two ways. One way is to run the Fast Euclidean Algorithm for the pair of relatively prime numbers σ and p, as in Worked Exercise 1.2. But there is a second way of doing this: Bob

knows α and, using his private key b, he can evaluate α^{p-1-b}. With his insight into the communication scheme, and using $\xi^{p-1} = 1$, he obtains

$$\alpha^{p-1-b} = (\xi^a)^{p-1-b} = (\xi^{p-1})^a \xi^{-ab} = 1 \times \sigma^{-1} = \sigma^{-1}.$$

Similarly, Alice can determine σ^{-1} by simply computing β^{p-1-a}.

Encryption and decryption

The information exchange can now go both ways as follows. Suppose that Alice wants to send Bob a message coded as $m \in GF_p$. She computes the product $c = m\sigma$ in GF_p, and transmits the ciphertext c through the channel to Bob. To decrypt, Bob just needs to multiply the ciphertext c by σ^{-1} to obtain the original message m. Bob uses the same encryption technique if he wants to send a message to Alice.

Example 1.14 *Diffie–Hellman encryption and decryption*

Suppose again that Alice wants to send Bob her three-digit credit card security number $m = 196$. For the sake of this demonstration example, they agree to use the prime $p = 257$.

The fact that $257 = 2^8 + 1$ enables Alice and Bob to quickly determine a primitive element of GF_{257}: they see right away that

$$2^8 \equiv -1 \pmod{257},$$

so that 2 cannot be a primitive element. How about 3? As $3^4 = 81$, Alice and Bob quickly check that, in GF_{257}, $3^8 = 136$ and $3^{16} = -8 = -2^3$. Further, in GF_{257} they also have $3^{32} = 2^6$, $3^{64} = 2^{12} = -2^4$, and finally $3^{128} = 2^8 = -1$. This means that 3 is a primitive element, since $3^{256} = 1$, and the fact that $3^{128} = -1$ proves that the multiplicative order of 3 in GF_{257} is indeed 256.

Alice and Bob can now choose as their ξ any power 3^j where j is relatively prime to 256, and they choose 3^{29}. To evaluate this power in GF_{257}, which is equivalent to evaluating $3^{29} \pmod{257}$, they use our well-known trick of writing the exponent as a sum of powers of 2:

$$3^{29} = 3^{16} \times 3^8 \times 3^4 \times 3 \equiv -8 \times 136 \times 81 \times 3 \equiv 69 \pmod{257},$$

and so they set $\xi = 69$.

Again, just to simplify calculations, Alice and Bob choose $a = 13$ and $b = 17$ as their private keys respectively. Alice computes the element $\alpha = \xi^a = 69^{13} = 51$ in GF_{257} and sends α to Bob, while Bob computes $\beta = \xi^b = 69^{17} = 163$ and sends this value to Alice. Then Alice determines

$$\sigma = \beta^a = 163^{13} = 163^8 \times 163^4 \times 163 = 151,$$

and Bob computes

$$\alpha^b = 51^{17} = 51^{16} \times 51 = 151,$$

all in GF_{257}. From our discussion of the communication setup, we know that they must obtain the same value, namely, $\sigma = 151$.

Again from our previous discussion, we know how both Alice and Bob can evaluate σ^{-1}. One way would be to determine the multiplicative inverse of 151 modulo 257 using the Fast Euclidean Algorithm; we did this in Worked Exercise 1.2. To use the second possibility outlined earlier, say from Bob's perspective, in GF_{257} he can compute

$$\sigma^{-1} = \alpha^{p-1-b} = 51^{239} = 51^{128+64+32+8+4+2+1} = 80,$$

which agrees with the solution of Worked Exercise 1.2.

Recall that Alice's message was $m = 196$. In GF_{257}, she computes

$$c = m \times \sigma = 196 \times 151 = 41$$

and sends this number to Bob, who decrypts the message c by simply working out $c \times \sigma^{-1} = 41 \times 80 = 196$ in GF_{257}.

Exercise 1.9

Alice and Bob want to exchange some information using the Diffie–Hellman cryptosystem with $p = 193$, $\xi = 5$. Alice chooses $a = 12$ as her private key and Bob chooses $b = 17$ as his. If Alice's message is $m = 123$, determine the ciphertext c and verify that Bob's decryption of the ciphertext gives back Alice's message.

Security

In order to ensure security in a Diffie–Hellman exchange, one would have to choose an enormously large prime number p to begin with. An analysis of the encryption and decryption algorithms then shows that if there was an 'easy' way to decrypt messages, then there would be an 'easy' way to determine Alice's and Bob's secret exponents a and b. If this was the case, there would be an 'easy' way to determine the exponent a from the publicly known values of ξ and $\alpha = \xi^a$. If α and ξ were positive real numbers, then the value of a would simply be equal to $\log_\xi \alpha$. However, it is far from clear how to compute a when α and ξ are elements of the finite field GF_p. It turns out that we may still think of a as a generalisation of a logarithm, for which we give the following definition.

Definition 1.15 *Discrete logarithm in finite fields*

Let $\alpha \neq 0$ and $\beta \neq 0, 1$ be elements of a finite field GF_p. If there exists an integer a such that $0 \leq a < p - 1$ and $\beta^a = \alpha$, then a is called the **discrete logarithm of α to the base β**. In such a case we write $a = \log_\beta \alpha$.

Observe that if $\beta = \xi$, a primitive element of GF_p, then the discrete logarithm always exists and is uniquely determined. So determining the exponent a in the previous paragraph is equivalent to determining the discrete logarithm $a = \log_\xi \alpha$.

However, it turns out that computing discrete logarithms in the field GF_p appears infeasible for very large primes p (at least several hundreds of decimal digits long) in real time. The lack of fast algorithms for computing discrete logarithms thus provides assurance that, in general, Diffie–Hellman encryption can be considered safe if the prime p and the primitive element ξ are chosen properly (again, going into more detail is well beyond our scope here).

The Diffie–Hellman system is rarely used in practice, mostly due to commercial reasons. The dominant public-key cryptosystems are based on RSA and, increasingly, on elliptic curves, which you will meet in the next section. We have included the Diffie–Hellman system in this chapter to present a cryptosystem based on the apparent infeasibility of the discrete logarithm problem. A generalisation of this problem is essential for the security of cryptosystems based on elliptic curves, which we are about to describe.

2 Cryptosystems based on elliptic curves

Under certain mild restrictions, an elliptic curve over a field F with constants $a, b \in F$ can be identified with the graph of the implicit function given by the equation $y^2 = x^3 + ax + b$, that is, with the set of all ordered pairs $(x, y) \in F \times F$ that satisfy the equation. If the underlying field F is the field of rational numbers, the corresponding elliptic curves play a decisive role in the proof of the famous Fermat's Last Theorem. Surprisingly, elliptic curves over a finite field have found non-trivial applications in public-key cryptography; such applications were suggested independently by Neal Koblitz and Victor S. Miller in 1985.

In this section we present a light version of the mathematical background needed to understand elliptic curves over various fields. Subsection 2.1 contains basic facts about cubic polynomials. Elliptic curves are formally defined in Subsection 2.2, with a number of examples. The striking feature of elliptic curves is that they give rise to an abelian group in a natural way, which we introduce in Subsection 2.3 and illustrate through some examples in Subsection 2.4. Finally, in Subsection 2.5 we introduce two cryptosystems based on elliptic curves over finite fields, the ElGamal and the Menezes–Vanstone systems. We also include a discussion on the security of these systems, which is based on an analogue of the discrete logarithm problem for elliptic curves.

2.1 First things first: cubic polynomials

Let F be a field. At this point we make no assumptions about F: it can be finite or infinite, although later we will focus on finite fields. As we shall

soon see, elliptic curves are defined by means of cubic polynomials over a field. Therefore we start by looking at such polynomials in some detail.

A general cubic polynomial over F with leading term equal to 1 has the form

$$z^3 + Az^2 + Bz + C, \quad \text{where} \quad A, B, C \in F. \tag{1}$$

If the characteristic of the field is not equal to 3, then the element $3 = 1 + 1 + 1 \in F$ is not equal to zero and hence has a multiplicative inverse in F, which we denote by $\frac{1}{3}$ as usual. In this case, we may make the substitution $z = x - \frac{1}{3}A$ in (1), which gives

$$\left(x - \frac{1}{3}A\right)^3 + A\left(x - \frac{1}{3}A\right)^2 + B\left(x - \frac{1}{3}A\right) + C. \tag{2}$$

Evaluating the powers and simplifying, turn the polynomial (2) into

$$x^3 + ax + b, \quad \text{where} \quad a = -\frac{1}{3}A^2 + B, \quad b = \frac{2}{27}A^3 - \frac{1}{3}AB + C. \tag{3}$$

Expression (3) tells us that if the characteristic of the field is not 3, we may simplify a cubic polynomial with leading term 1 by removing the term containing the square of the indeterminate. Therefore from now on we will only consider cubic polynomials over a field F of the form

$$f(x) = x^3 + ax + b, \quad a, b \in F. \tag{4}$$

In doing so, we will take into the account the loss of generality when the characteristic of the field is equal to 3 – no loss of generality occurs for fields of characteristic distinct from 3. In addition, we will restrict the discussion to the case when $f(x)$ has no multiple roots. We delay an explanation of the reason for this restriction until after Proposition 2.14 in Subsection 2.3.

In the terminology introduced at the end of Book B, Chapter 7, Subsection 4.2, cubic equations (that is, polynomial equations of degree 3) can be solved by radicals. A formula for the solution was discovered by the Italian mathematician Gerolamo Cardano in medieval times, and it is far more complex than the formula for solving quadratic equations. However, a sufficient condition for a cubic polynomial of the form $x^3 + ax + b$ to have no multiple root can be obtained surprisingly easily, and without invoking Cardano's formula.

Lemma 2.1

Let $f(x) = x^3 + ax + b$ be a polynomial over an arbitrary field F. If $4a^3 + 27b^2 \neq 0$, then $f(x)$ has no multiple roots.

You are asked to prove Lemma 2.1 in the next exercise. Observe that we do not assume that the characteristic of the field is different from 2 and 3, so even in such fields $4a^3 + 27b^2 \neq 0$ is a sufficient condition for $f(x) = x^3 + ax + b$ to have no multiple roots, but we will not need this fact.

Exercise 2.1 *Proof of Lemma 2.1*

Prove Lemma 2.1.

Hint: try proving the contrapositive.

In the next section we will need more information about the behaviour of the graphs of cubic polynomials over the field $\mathbb{R}$ of real numbers. To derive this information, we will recall how to sketch the graph of $f(x)$ in the real plane with the help of the first and the second derivative. You are likely to have met this method to graph real functions in any calculus courses in your previous mathematical studies.

Thus, let $f(x) = x^3 + ax + b$ be such that the coefficients a, b and the values of f are real numbers. The derivative $f'(x) = 3x^2 + a$ has no real roots if $a > 0$, and two real roots if $a < 0$. It follows that $f(x)$ is increasing everywhere if $a > 0$, and its graph has a shape as in Figure 2.1.

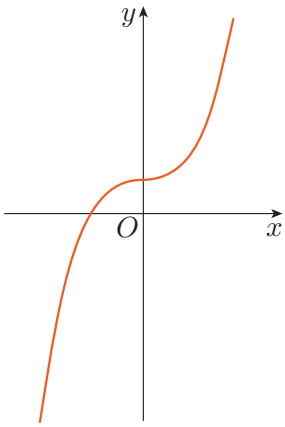
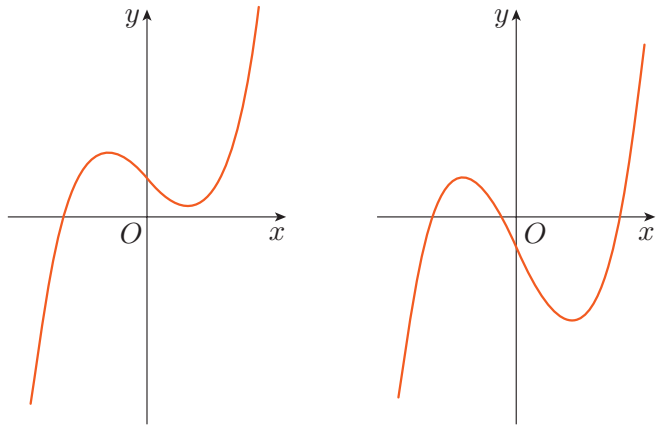


Figure 2.1 The shape of the graph of $f(x) = x^3 + ax + b$ for $a > 0$

If $a < 0$, then $f(x)$ is increasing in the intervals $(-\infty, -\sqrt{-\frac{a}{3}})$ and $(\sqrt{-\frac{a}{3}}, \infty)$ and decreasing in the interval $(-\sqrt{-\frac{a}{3}}, \sqrt{-\frac{a}{3}})$, with a local maximum at $x = -\sqrt{-\frac{a}{3}}$ and a local minimum at $x = +\sqrt{-\frac{a}{3}}$. The second derivative $f''(x) = 6x$ shows that the graph of $f(x)$ is concave down in the interval $(-\infty, 0)$ and concave up in $(0, \infty)$. Since the values of $f(x)$ approach $+\infty$ and $-\infty$ as x tends to $+\infty$ and $-\infty$, respectively, the shape of the cubic function $f(x)$ for $a < 0$ is either as in Figure 2.2(a) or as in Figure 2.2(b).



(a) (b)

Figure 2.2 Possible shapes for the graph of $f(x) = x^3 + ax + b$ for $a < 0$

We now state and prove a necessary and sufficient condition for the polynomial $f(x) = x^3 + ax + b$ over the field $\mathbb{R}$ with $4a^3 + 27b^2 \neq 0$ to have three distinct real roots.

Lemma 2.2

Let $f(x) = x^3 + ax + b$ be a cubic polynomial over $\mathbb{R}$ such that $4a^3 + 27b^2 \neq 0$. Then $f(x)$ has three real roots if $a < 0$ and $|b| < \frac{2|a|\sqrt{|a|}}{3\sqrt{3}}$, and exactly one real root otherwise.

Proof From calculus and the shape of the graph of $f(x)$ we know that the polynomial $f(x)$ over $\mathbb{R}$ has a double root if, and only if, the x-axis is a tangent line to the graph of the cubic function $f(x)$, and then the double root is precisely the point of tangency. Also, Figure 2.2 and the discussion preceding it show that if $f(x)$ has a double root, then $a < 0$ and the double root must be the point x_0 at which $f(x)$ attains its local maximum or its local minimum. This happens precisely at the point x_0 for which $f'(x_0) = 0$, that is, at $x_0 = \pm\sqrt{\frac{|a|}{3}}$ for $a < 0$. By the proof of Lemma 2.1 we know that in such a case we have $b = 2x_0^3$, or, equivalently, $b = \pm\frac{2|a|\sqrt{|a|}}{3\sqrt{3}}$.

Since the value of b represents a vertical shift of the graph of $x^3 + ax$ depicted in Figure 2.3, we conclude that $f(x)$ has three real roots if, and only if, $a < 0$ and the shift b lies in the interval $(-\frac{2|a|\sqrt{|a|}}{3\sqrt{3}}, \frac{2|a|\sqrt{|a|}}{3\sqrt{3}})$. ■

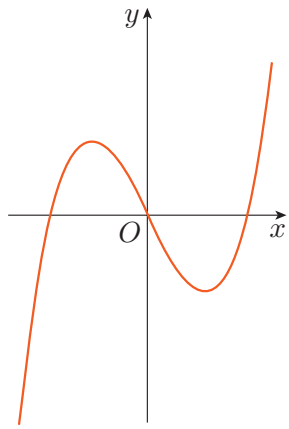


Figure 2.3 The shape of the graph of $x^3 + ax$ for $a < 0$

2.2 Elliptic curves: definition and examples

We are now ready to state one of the central definitions of this chapter.

Definition 2.3 *Elliptic curve*

Let F be a field of characteristic distinct from 2 and 3, and let $f(x) = x^3 + ax + b$ be a cubic polynomial over F such that $4a^3 + 27b^2 \neq 0$. Let $\mathcal{O}$ be a symbol, distinct from all elements of $F \times F$. The set

$$E_F(a, b) = \{\mathcal{O}\} \cup \{(x, y) \in F \times F : \ y^2 = f(x)\}$$

is called an **elliptic curve** over the field F.

Elements of an elliptic curve $E_F(a, b)$ are usually referred to as **points** on the curve. The special symbol $\mathcal{O}$ is sometimes called a **point at infinity**. At this time the symbol $\mathcal{O}$ may appear artificial; its importance will be made clear later.

Observe that if $P = (x, y)$ is a point on an elliptic curve $E_F(a, b)$, then the point with coordinates $(x, -y)$, denoted $-P$, is also a point on the same curve. Thus, P and $-P$ have the same x-coordinate but their y-coordinates differ in sign. In the special case when $y = 0$, that is, when $P = (x, 0)$ and $x \in F$ is a (simple) root of the polynomial $f(x) = x^3 + ax + b$, we have $P = -P$. If we set $\mathcal{O} = -\mathcal{O}$, the map $P \mapsto -P$ defines a bijection of $E_F(a, b)$ into itself.

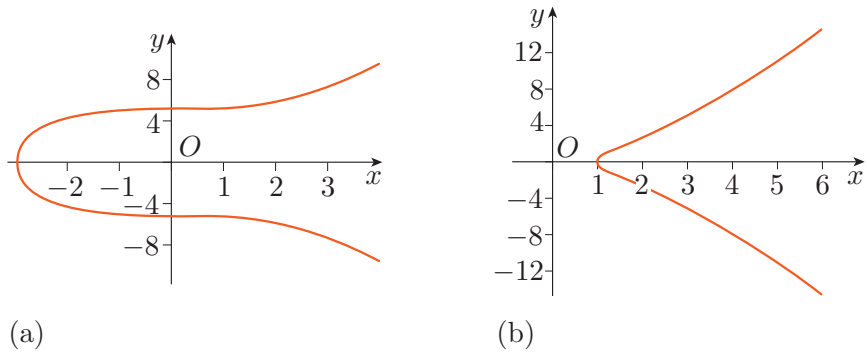
We now look at elliptic curves over a number of different fields and we discuss the possible shapes of their graphs.

Example 2.4 *Real elliptic curves*

Let $f(x) = x^3 + b$ be a polynomial over the field of real numbers, where b is a non-zero real number. The polynomial $f(x)$ has a single real root, $x_0 = \sqrt[3]{-b}$. Moreover, the square root $\sqrt{f(x)}$ is real if, and only if, $x^3 + b \geq 0$, that is, if, and only if, $x \geq x_0$. Thus, the elliptic curve $E_{\mathbb{R}}(0, b)$ contains the point $(x_0, 0)$ and, for every $x > x_0$, the pair of points $P_x = (x, \sqrt{f(x)})$ and $-P_x = (x, -\sqrt{f(x)})$, together with $\mathcal{O}$. Formally,

$$E_{\mathbb{R}}(0, b) = \{\mathcal{O}\} \cup \{(x, \pm\sqrt{x^3 + b}) : \ x \in \mathbb{R}, \ x \geq \sqrt[3]{-b}\}.$$

The graph of this kind of elliptic curve is shaped either as in the example in Figure 2.4(a) or the example in Figure 2.4(b).



(a) (b)

Figure 2.4 The elliptic curves: (a) $E_{\mathbb{R}}(0, 27)$, that is, $y^2 = x^3 + 27$; (b) $E_{\mathbb{R}}(0, -1)$, that is, $y^2 = x^3 - 1$

Inspired by Example 2.4, we will now attempt to study a general elliptic curve $E_{\mathbb{R}}(a, b) = \{\mathcal{O}\} \cup \{(x, y) \in \mathbb{R} \times \mathbb{R} : \ y^2 = x^3 + ax + b\}$ corresponding to a polynomial $f(x) = x^3 + ax + b$ over $\mathbb{R}$, with $4a^3 + 27b^2 \neq 0$ and $a \neq 0$. Note that, up to the point $\mathcal{O}$, graphing $E_{\mathbb{R}}(a, b)$ is equivalent to graphing the curve $|y| = \sqrt{x^3 + ax + b}$.

If $f(x)$ has exactly one real root x_0, then the domain of the function $y = \sqrt{x^3 + ax + b}$ is the interval $[x_0, \infty)$. The graph of $y = \sqrt{x^3 + ax + b}$, and hence the graph of $E_{\mathbb{R}}(a, b)\backslash\{\mathcal{O}\}$, in the open interval (x_0, ∞) is obtained by taking $\pm$ the square root of the values on the part of the graph of $f(x)$ that lies above the x-axis.

Let us now discuss the case when $f(x) = x^3 + ax + b$ with $4a^3 + 27b^2 \neq 0$ has exactly 3 real roots, say, $x_1 < x_2 < x_3$. By Lemma 2.2, this happens if, and only if, $a < 0$ and $|b| < \frac{2|a|\sqrt{|a|}}{3\sqrt{3}}$. In this case the domain of the curve $|y| = \sqrt{x^3 + ax + b}$ is $[x_1, x_2] \cup [x_3, \infty)$ and the curve has two components. Its graph is again obtained by taking $\pm$ the square root of the values on the two parts of the graph of $f(x)$ that lie above the x-axis.

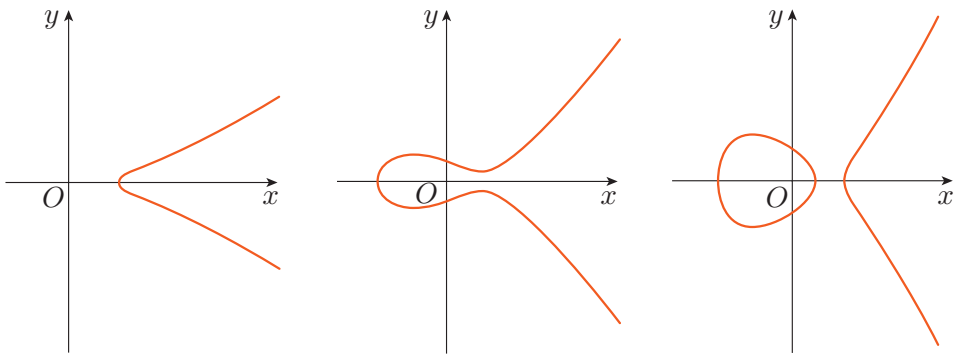
The three typical shapes of elliptic curves $E_{\mathbb{R}}(a,b)$ are depicted in Figures 2.5(a), 2.5(b) and 2.5(c) in the cases

(i) $b < 0 < a$,

(ii) $a < 0$ and $b > \frac{2|a|\sqrt{|a|}}{3\sqrt{3}}$, and

(iii) $a < 0$ and $|b| < \frac{2|a|\sqrt{|a|}}{3\sqrt{3}}$.

In all these situations, observe that the bijection $P \mapsto -P$ of any elliptic curve over $\mathbb{R}$ into itself shows as a reflection of the graph in the x-axis.



(a) (b) (c)

Figure 2.5 The shape of the graph of an elliptic curve $E_{\mathbb{R}}(a,b)$ where: (a) $b < 0 < a$; (b) $a < 0$ and $b > \frac{2|a|\sqrt{|a|}}{3\sqrt{3}}$; (c) $a < 0$ and $|b| < \frac{2|a|\sqrt{|a|}}{3\sqrt{3}}$

Example 2.5 *Complex elliptic curves*

Let $\mathbb{C}$ denote the field of complex numbers, and let $f(x) = x^3 + b$, where $b \neq 0$. What does the corresponding elliptic curve $E_{\mathbb{C}}(0,b)$ look like?

Let $\omega = e^{\frac{2\pi i}{3}} = -\frac{1}{2} + i\frac{\sqrt{3}}{2}$. Then the three roots of $f(x)$ are

$$x_j = \omega^j \sqrt[3]{-b} \text{ for } j = 0, 1, 2.$$

It follows that $E_{\mathbb{C}}(0,b)$ contains the three points $(x_j, 0)$ for $j = 0, 1, 2$. For $x \in \mathbb{C}$ such that $x \notin \{x_0, x_1, x_2\}$, we have that $f(x) \neq 0$ and the equation $y^2 = f(x)$ has two distinct complex roots. If we denote one of the roots by $\sqrt{f(x)}$, then every $x \in \mathbb{C}\backslash\{x_0, x_1, x_2\}$ gives rise to two distinct points $(x, \sqrt{f(x)})$ and $(x, -\sqrt{f(x)})$ in $E_{\mathbb{C}}(0,b)$. In summary,

$$E_{\mathbb{C}}(0,b) = \{\mathcal{O}\} \cup \{(x, \pm\sqrt{x^3 + b}) : x \in \mathbb{C}\}.$$

Obtaining a graph of $E_{\mathbb{C}}(0,b)$ is a difficult task. Nevertheless, more advanced methods than the ones at our disposal show that $E_{\mathbb{C}}(0,b)$ can be represented in $\mathbb{R}^3$ as the set of points of a torus, an object that looks like the inner tube of a tyre.

Example 2.6 *Rational elliptic curves*

We now consider elliptic curves over the field $\mathbb{Q}$ of rational numbers. Let $f(x) = x^3 + ax + b$ be such that $a, b \in \mathbb{Q}$ and $4a^3 + 27b^2 \neq 0$. Besides the exceptional point $\mathcal{O}$, an elliptic curve $E_{\mathbb{Q}}(a, b)$ consists of ordered pairs (x, y) of rational numbers such that $y^2 = x^3 + ax + b$. This means that for (x, y) to be a point on $E_{\mathbb{Q}}(a, b)$ for some rational number x, the square root $y = \sqrt{x^3 + ax + b}$ must be a rational number as well.

This is a very restrictive requirement and its full understanding is one of the major open problems in contemporary mathematics. To illustrate the difficulties, we list four examples, on which we comment below:

(i) $E_{\mathbb{Q}}(1, -47)$ consists just of the point at infinity, $\mathcal{O}$

(ii) $E_{\mathbb{Q}}(1, 2) = \{\mathcal{O}, (-1, 0), (1, 2), (1, -2)\}$

(iii) $E_{\mathbb{Q}}(1, 1)$ is infinite and can be 'generated' by one point

(iv) $E_{\mathbb{Q}}(1, 9)$ is infinite and can be 'generated' by two points but not by one.

Example (i) tells us that the equation $y^2 = x^3 + x - 47$ has no rational solution, while example (ii) tells us that the equation $y^2 = x^3 + x + 2$ has just 3 rational solutions.

Example (iii) is more exciting, as the equation $y^2 = x^3 + x + 1$ has an infinite number of rational solutions. The surprise, however, comes from the first 14 rational solutions (x, y) in their simplest form:

$$(0, \pm 1)\,,\ (72, \pm 611)\,,\ \left(\frac{1}{4}, \pm\frac{9}{8}\right),\ \left(\frac{-287}{1296}, \pm\frac{40\,879}{46\,656}\right),\ \left(\frac{43\,992}{82\,369}, \pm\frac{30\,699\,397}{23\,639\,903}\right),$$
$$\left(\frac{26\,862\,913}{1\,493\,284}, \pm\frac{139\,455\,877\,527}{1\,824\,793\,048}\right),\ \left(-\frac{3\,596\,697\,936}{8\,760\,772\,801}, \pm\frac{591\,456\,591\,665\,497}{819\,999\,573\,400\,799}\right).$$

The sense in which all the points on the rational elliptic curve $E_{\mathbb{Q}}(1, 1)$ are 'generated' by a single point will be explained later: we will see that every elliptic curve is, at the same time, an abelian group with an operation of addition, which is derived from certain geometric properties of the graphs of these curves. Moreover, the abelian group defined on $E_{\mathbb{Q}}(1, 1)$ turns out to be isomorphic to the additive group of the integers, $(\mathbb{Z}, +)$, which, as you know, is generated by the single element 1.

In contrast with this, the elliptic curve $E_{\mathbb{Q}}(1, 9)$ in example (iv) turns out to be isomorphic, as an abelian group, to the direct product $\mathbb{Z} \times \mathbb{Z}$. This group can be generated by two generators, but not by one. Complete proofs of these facts are well beyond the scope of this module.

Exercise 2.2

Find all the points (x, y) on the rational elliptic curve $E_{\mathbb{Q}}(1, 9)$ such that x, y are integers and $0 \leq x \leq 10$.

This section concludes with an example of a family of elliptic curves over the finite field GF_p where p is any odd prime such that $p \equiv 2 \pmod 3$.

Example 2.7 *Elliptic curves over finite fields*

Let p be an arbitrary odd prime such that $p \equiv 2 \pmod 3$ and let GF_p be the finite field of order p. For an arbitrary non-zero $b \in GF_p$ we consider the elliptic curve $E_{GF_p}(0, b)$ defined by the cubic polynomial $f(x) = x^3 + b$ over GF_p.

Recall that the multiplicative group GF_p^* of GF_p has order $p - 1$ and it is cyclic. Our assumption on p implies that $p - 1$ is not a multiple of 3, which in turn implies that GF_p^* contains no element of order 3. We prove that in this case for any $c \in GF_p$ the equation $z^3 = c$ has a unique solution $z \in GF_p$.

This statement is true for $c = 0$. To prove it for all $c \neq 0$, it is sufficient to show that the map $z \mapsto z^3$ is a bijection of the set GF_p^* into itself. Since GF_p^* is finite, it is sufficient to show that the map $z \mapsto z^3$ is injective on GF_p^*. Assume that $z_1^3 = z_2^3$ for some $z_1, z_2 \in GF_p^*$. Then $(z_2^{-1} z_1)^3 = 1$, and since GF_p^* contains no element of order 3, we must have $z_2^{-1} z_1 = 1$, and hence $z_2 = z_1$.

Any injective map of a finite set into itself is a bijection.

Thus, if p is odd and $p \equiv 2 \pmod 3$, for every $c \in GF_p$ there is a unique $z \in GF_p$ such that $z^3 = c$; we will denote this element z by $\sqrt[3]{c}$ as usual.

We can now determine the elliptic curve $E_{GF_p}(0, b)$ by listing all ordered pairs (x, y) of elements of GF_p such that $y^2 = x^3 + b$.

Firstly, the point $(\sqrt[3]{-b}, 0)$ is on the curve. Secondly, for any $x \in GF_p$ such that $x \neq \sqrt[3]{-b}$, the map $\theta\colon\ x \mapsto x^3 + b$ is, by a similar argument to that for $z \mapsto z^3$, a bijection from the set $GF_p \backslash \{\sqrt[3]{-b}\}$ onto GF_p^*. In order to determine y, we must be able to take the square root of $x^3 + b$, and this can be done if, and only if, $x^3 + b$ is a square in GF_p^*.

So let S denote the set of all the elements $x \in GF_p \backslash \{\sqrt[3]{-b}\}$ such that $x^3 + b$ is a square in GF_p^*. Since the map θ is a bijection and GF_p^* contains exactly $\frac{p-1}{2}$ squares by Proposition 1.4 of Book C, Chapter 10, we have $|S| = \frac{p-1}{2}$. Finally, for each $x \in S$ there are two values of the square root of $x^3 + b$, for which we use the customary notation $\pm\sqrt{x^3 + b}$. It follows that, for any odd prime $p \equiv 2 \pmod 3$ and any $b \in GF_p^*$ we have

$$E_{GF_p}(0, b) = \{\mathcal{O}\} \cup \{(\sqrt[3]{-b}, 0)\} \cup \{(x, \pm\sqrt{x^3 + b})\colon\ x \in S\}\,;$$

in particular, any such elliptic curve contains exactly $p + 1$ points.

Elliptic curves of the kind considered in Example 2.7 can be graphed, but since the curves are finite, their graphs appear as finite collections of isolated points.

Consider for instance the elliptic curve $E_{GF_{11}}(0,1)$ without the point $\mathcal{O}$. For our calculations, let us identify GF_{11} with the set $\{0, \pm1, \pm2, \pm3, \pm4, \pm5\}$. To follow the method outlined in Example 2.7, we find that $\sqrt[3]{-1} = -1$, and then we work out the set of squares in GF_{11}^* and the values of $x^3 + 1$ as x ranges over the elements of $GF_{11} \setminus \{-1\}$.

Table 2.1 The squares in GF_{11}^*

y	±1	±2	±3	±4	±5
y^2	1	4	-2	5	3

Table 2.2 The values of $x^3 + 1$ in $GF_{11} \setminus \{-1\}$

x	0	1	2	3	4	5	$6 \equiv -5$	$7 \equiv -4$	$8 \equiv -3$	$9 \equiv -2$
$x^3 + 1$	**1**	2	**-2**	-5	-1	**5**	-3	**3**	-4	**4**

Table 2.1 shows that the set of squares in GF_{11}^* is $\{1, -2, 3, 4, 5\}$. In Table 2.2, the elements that are squares have been highlighted in blue. We infer that $S = \{0, 2, -2, -4, 5\}$. Therefore

$$E_{GF_{11}}(0,1) = \{\mathcal{O}, (0, \pm1), (2, \pm3), (5, \pm4), (7, \pm5), (9, \pm2), (10, 0)\}.$$

A graph of $E_{GF(11)}(0,1)\setminus\{\mathcal{O}\}$ is in Figure 2.6.

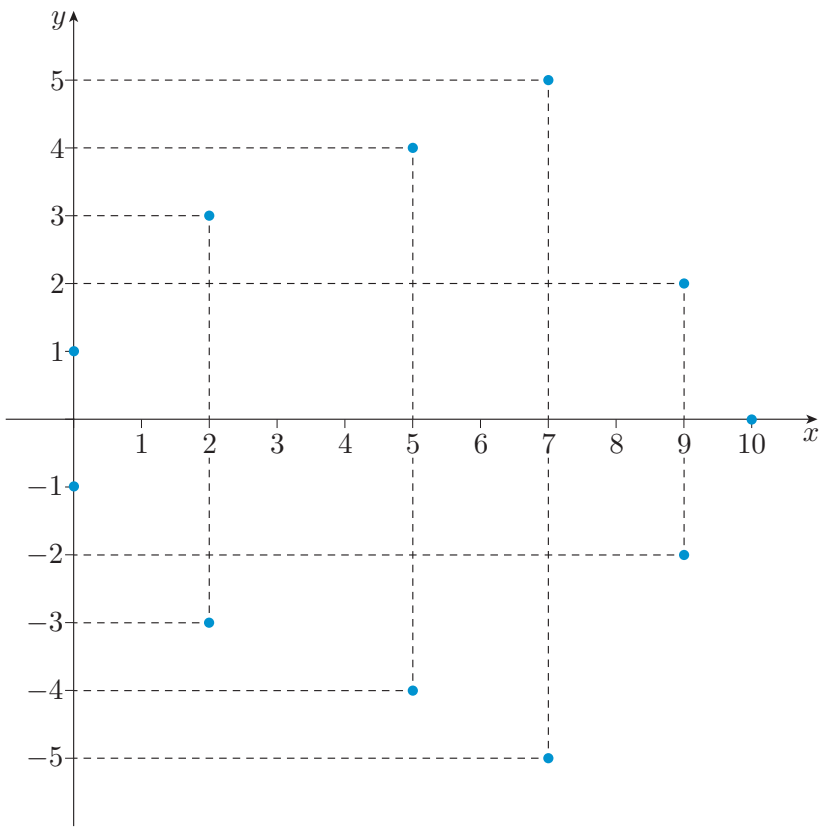


Figure 2.6 A graph of the elliptic curve $E_{GF_{11}}(0,1)\setminus\{\mathcal{O}\}$

Exercise 2.3

Draw the elliptic curve $E_{GF_{13}}(0, 2)\backslash\{\mathcal{O}\}$.

2.3 Elliptic curves as abelian groups

We now introduce an operation on the points of an elliptic curve that makes the curve into an abelian group. In the case of elliptic curves over the field of real numbers, the operation can be defined in geometric terms by considering (straight) lines that intersect the curve. In order to generalise this idea to arbitrary fields, we first generalise the notion of a line over a field.

Definition 2.8 *Line with slope*

Let F be a field and let $\mathcal{O} \notin F \times F$ be the point at infinity.

(a) For any chosen $c, d \in F$, a **line with slope** c is the set

$$L(c, d) = \{(x, y) \in F \times F : y = cx + d\}.$$

(b) For any $x_0 \in F$, a **slopeless line** is the set

$$L(x_0) = \{\mathcal{O}\} \cup \{(x, y) \in F \times F : x = x_0\}.$$

To simplify the terminology, the term **line over** F will be used to refer to both lines with slope and slopeless lines.

Lines with slope are a generalisation of the real lines $y = cx + d$. Slopeless lines over arbitrary fields are also generalisations of the real lines $x = x_0$ parallel to the y-axis, except that we have defined slopeless lines to include $\mathcal{O}$, our mysterious point at infinity. In particular, by definition, any two distinct slopeless lines intersect at exactly one point, namely, $\mathcal{O}$.

We can now also generalise the notion of a tangent line to an elliptic curve over a field. We take inspiration in the field $\mathbb{R}$ again.

Suppose that $y^2 = x^3 + ax + b$ defines an elliptic curve $E_{\mathbb{R}}(a, b)$ over $\mathbb{R}$ and let $P = (x_0, y_0)$ be a point on the graph of this curve. The slope of the tangent line to the curve at P is found by implicit differentiation, which you will have met in your previous studies. Differentiating $y^2 = x^3 + ax + b$ with respect to x gives

$$2yy'(x) = 3x^2 + a,$$

and evaluating at P gives

$$2y_0y'(x_0) = 3x_0^2 + a. \tag{5}$$

It follows that the slope of the tangent line to our curve at P is undefined if $y_0 = 0$. If $y_0 \neq 0$, however, the slope c_0 of the tangent line is equal to $y'(x_0)$ calculated from (5), that is,

$$c_0 = \frac{3x_0^2 + a}{2y_0}. \tag{6}$$

Thus, if $y_0 \neq 0$, the equation of the tangent line at P is $y - y_0 = c_0(x - x_0)$, or

$$y = c_0 x + d_0 \text{ where } d_0 = y_0 - c_0 x_0. \tag{7}$$

Substituting for c_0 from (6) into the equation for d_0 in (7), using $y_0^2 = x_0^3 + ax_0 + b$ and simplifying give

$$d_0 = \frac{-x_0^3 + ax_0 + 2b}{2y_0}. \tag{8}$$

If $y_0 = 0$, the tangent line at P is slopeless and has equation $x = x_0$. Equivalently, $L(c_0, d_0)$, where c_0 and d_0 are as in (6) and (8) respectively, is the tangent line to the elliptic curve $E_{\mathbb{R}}(a, b)$ at the point (x_0, y_0) if $y_0 \neq 0$, and $L(x_0)$ is the tangent to $E_{\mathbb{R}}(a, b)$ at the point (x_0, y_0) if $y_0 = 0$.

All these considerations hold for elliptic curves over the field $\mathbb{R}$ of real numbers. We extend these ideas to arbitrary fields in the next definition.

Definition 2.9 *Tangent line with slope*

Let $E_F(a, b)$ be an elliptic curve over a field F of characteristic distinct from 2 and 3, and let $P = (x_0, y_0)$ be a point on this curve.

(a) If $y_0 \neq 0$, the line $L(c_0, d_0)$, where

$$c_0 = \frac{3x_0^2 + a}{2y_0} \quad \text{and} \quad d_0 = \frac{-x_0^3 + ax_0 + 2b}{2y_0},$$

is said to be the **tangent line with slope** c_0 to the curve at P.

(b) If $y_0 = 0$, the line $L(x_0)$ is said to be the **slopeless tangent line** to $E_F(a, b)$ at P.

Again, to simplify the terminology, we will simply speak about **tangent lines** to an elliptic curve.

We now prove a basic auxiliary result relating elliptic curves to lines, which will enable us to introduce addition of points of an elliptic curve.

Lemma 2.10

Let $E_F(a, b)$ be an elliptic curve over a field F of characteristic distinct from 2 and 3.

(a) Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be points on $E_F(a, b)$ distinct from $\mathcal{O}$ and such that $P_1 \neq \pm P_2$. Then the (unique) line through P_1 and P_2 is $L(c, d)$, where

$$c = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{and} \quad d = \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1}.$$

This line intersects the elliptic curve at the point $\pi(P_1, P_2) = (x_3, y_3)$, where $x_3 = c^2 - x_1 - x_2$ and $y_3 = cx_3 + d$.

(b) Let $P_0 = (x_0, y_0)$ be a point on $E_F(a, b)$ such that $y_0 \neq 0$. Then the tangent line $L(c_0, d_0)$, where c_0 and d_0 are as in Definition 2.9, intersects the curve $E_F(a, b)$ in the point $\pi'(P_0) = (x', y')$, where $x' = c_0^2 - 2x_0$ and $y' = c_0 x' + d_0$.

Proof

(a) Let P_1 and P_2 be as in our statement. Since $x_1 \neq x_2$, the unique line through P_1 and P_2 over F has equation

$$y - y_1 = c(x - x_1)$$

exactly as in calculus, with slope

$$c = \frac{y_2 - y_1}{x_2 - x_1}.$$

Equivalently, the line has equation $y = cx + d$, where $d = y_1 - cx_1$. Substituting for c yields

$$d = \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1}.$$

To calculate any other intersection points of the line $L(c, d)$ of equation $y = cx + d$ with our elliptic curve of equation $y^2 = x^3 + ax + b$, we eliminate y, which gives the polynomial equation

$$(cx + d)^2 = x^3 + ax + b,$$

or, equivalently,

$$x^3 - c^2x^2 + (a - 2cd)x + (b - d^2) = 0.$$

This cubic equation has three roots (possibly repeated). By our assumption, however, we already know two distinct roots of this equation, namely, x_1 and x_2. If we denote the third root by x_3, we have

$$x^3 - c^2x^2 + (a - 2cd)x + (b - d^2) = (x - x_1)(x - x_2)(x - x_3). \quad (9)$$

Comparing the coefficients of x^2 on both sides of equation (9) gives $x_1 + x_2 + x_3 = c^2$ and hence $x_3 = c^2 - x_1 - x_2$. If $\pi(P_1, P_2) = (x_3, y_3)$ is the point of intersection of $L(c, d)$ and $E_F(a, b)$, and the x-coordinate is equal to x_3, then $y_3 = cx_3 + d$. This proves the first part of our lemma.

(b) By Definition 2.9, the tangent line to the curve at P_0 is $L(c_0, d_0)$, where

$$c_0 = \frac{3x_0^2 + a}{2y_0}, \quad d_0 = \frac{-x_0^3 + ax_0 + 2b}{2y_0}.$$

To calculate any other points of intersection of the line $L(c_0, d_0)$ with the elliptic curve, we again look for simultaneous solutions to the equations $y = c_0x + d_0$ and $y^2 = x^3 + ax + b$. Eliminating y yields

$$(c_0x + d_0)^2 = x^3 + ax + b,$$

or, equivalently,

$$x^3 - c_0^2x^2 + (a - 2c_0d_0)x + (b - d_0^2) = 0.$$

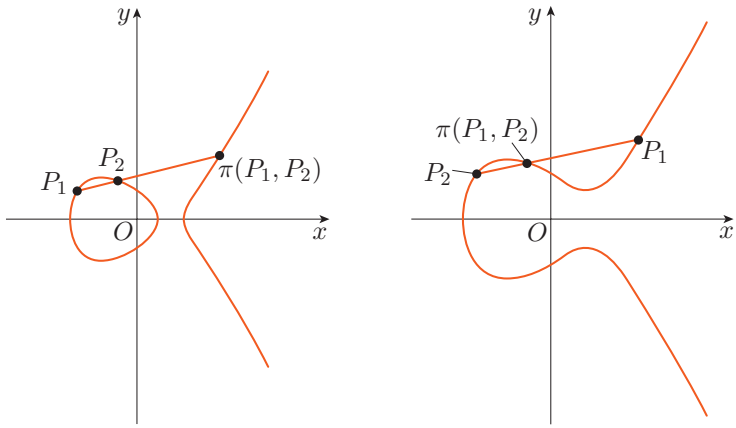
If the underlying field is the field of real numbers $\mathbb{R}$, a point of tangency is a root of the last equation of multiplicity at least two. This is also true over a finite field, although we shall not give a proof here. So, in general, since $L(c_0, d_0)$ is a tangent line at x_0, we know that x_0 is a root of multiplicity at least two. Thus if we denote the remaining root by x_3, we obtain

$$(x - x_0)^2(x - x_3) = x^3 - c_0^2x^2 + (a - 2c_0d_0)x + (b - d_0^2).$$

Comparing the coefficients of x^2 gives $x_3 = c_0^2 - 2x_0$, and so $y_3 = c_0x_3 + d_0$. The substitution $x_3 = x'$ and $y_3 = y'$ gives the required coordinates for the point of intersection. ■

We note that the points $\pi(P_1, P_2)$ and $\pi'(P_0)$ in Lemma 2.10 are not necessarily distinct from one of P_1, P_2 and from P_0, respectively.

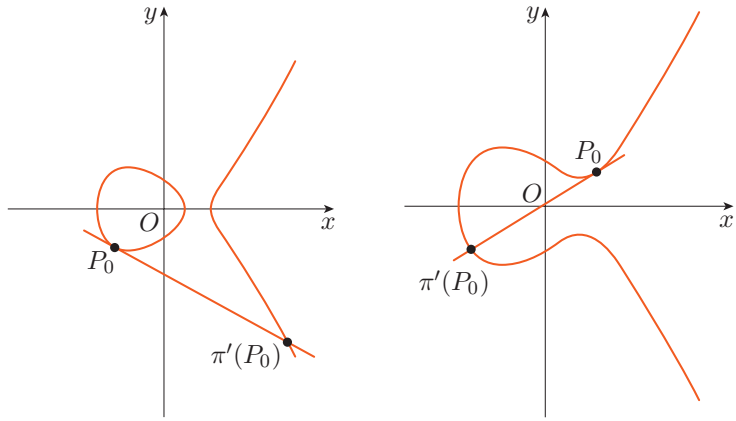
Two possible configurations for P_1, P_2 and $\pi(P_1, P_2)$ are depicted in Figure 2.7.



(a) (b)

Figure 2.7 Two possible configurations for P_1, P_2 and $\pi(P_1, P_2)$

Similarly, two typical situations that one may encounter in determining $\pi'(P_0)$ are given in Figure 2.8.



(a) (b)

Figure 2.8 Two possible configurations for P_0 and $\pi'(P_0)$

Worked Exercise 2.11 *Computing $\pi(P_1, P_2)$ and $\pi'(P_0)$*

Consider the elliptic curve $E_{GF_{19}}(1,1)$ corresponding to $y^2 = x^3 + x + 1$ over GF_{19}, together with the points $P_1 = (0,1)$, $P_2 = (5,6)$ and $P_0 = (9,-6)$ on this curve. Determine the points $\pi(P_1, P_2)$ and $\pi'(P_0)$.

Solution

To calculate $\pi(P_1, P_2)$ we use part (a) of Lemma 2.10 for the values $a = b = 1$, $(x_1, y_1) = (0,1)$ and $(x_2, y_2) = (5,6)$. We obtain

$$c = \frac{6-1}{5-0} = 1 \quad \text{and} \quad d = \frac{1 \times 5 - 6 \times 0}{5-0} = 1,$$

where the calculations are in GF_{19}, and

$$x_3 = 1^2 - 5 - 0 = -4, \quad y_3 = 1 \times (-4) + 1 = -3,$$

again in GF_{19}. It follows that $\pi(P_1, P_2) = (-4, -3)$.

Similarly, to compute $\pi'(P_0)$ we follow part (b) of Lemma 2.10 with $a = b = 1$ and $(x_0, y_0) = (9, -6)$. To determine c_0, we must evaluate the quotient

$$c_0 = \frac{3 \times 9^2 + 1}{2 \times (-6)} = \frac{244}{-12}$$

in GF_{19}, that is, we must solve the linear congruence

$$-12c_0 \equiv 244 \pmod{19}.$$

Since $244 \equiv 16 \pmod{19}$ and $-12 \equiv 7 \pmod{19}$, the strategy for solving linear congruences in Book A, Chapter 3, Subsection 3.3 tells us that this congruence is equivalent to

$$7c_0 \equiv 16 \pmod{19}.$$

Multiplying each side by 3 modulo 19 gives

$$2c_0 \equiv 10 \pmod{19},$$

which gives $c_0 = 5$. Similarly, finding

$$d_0 = \frac{-9^3 + 1 \times 9 + 2 \times 1}{2 \times (-6)} = \frac{-718}{-12} = \frac{359}{6}$$

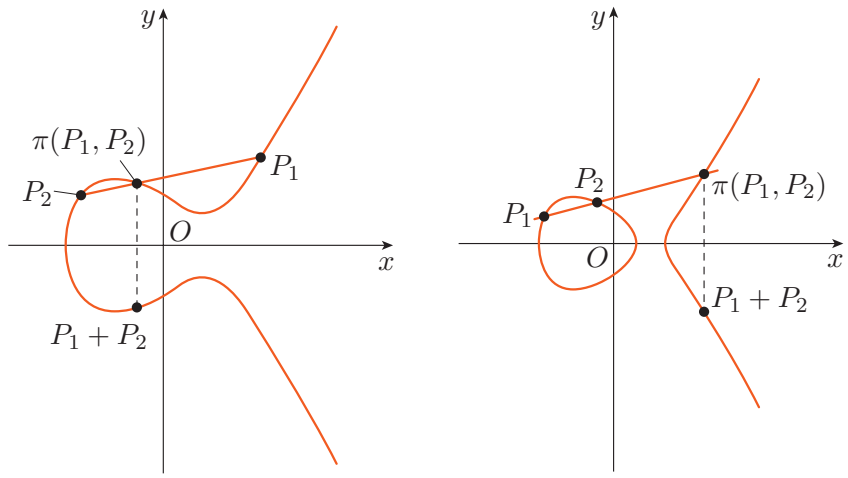
in GF_{19} is equivalent to solving the linear congruence

$$6d_0 \equiv 359 \pmod{19}.$$

Again, using the strategy given in Book A, Chapter 3, Subsection 3.3 gives $d_0 = 6$. We can now work out $x' = 5^2 - 2 \times 9 \equiv 7 \pmod{19}$ and $y' = 5 \times 7 + 6 \equiv 3 \pmod{19}$. Therefore $\pi'(P_0) = (7, 3)$.

Lemma 2.10 will enable us to define an operation of addition of points of an elliptic curve. To help intuition, we first give an informal description of the operation when the underlying field is the field of real numbers.

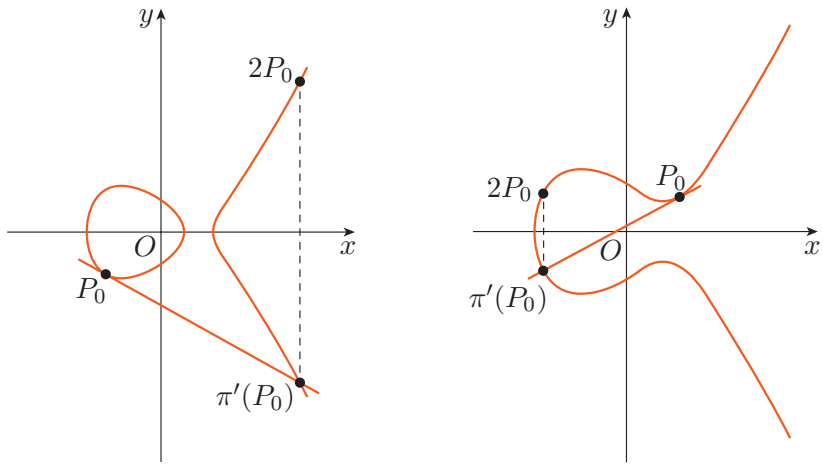
Suppose that P_1 and P_2 are two distinct points on an elliptic curve $E_{\mathbb{R}}(a, b)$ and suppose that the unique line through P_1 and P_2 intersects the curve in a third point $P_3 = \pi(P_1, P_2)$ distinct from P_1, P_2. We will then define $P_1 + P_2$ to be equal to $-P_3$. Two examples are illustrated in Figure 2.9.



(a) (b)

Figure 2.9 Two examples of addition on an elliptic curve

If P_0 is a point on $E_{\mathbb{R}}(a, b)$ such that the tangent line to the curve at P_0 is not vertical and intersects the curve in a point $\pi'(P_0)$ distinct from P_0, then we define $P_0 + P_0 = -\pi'(P_0)$. Two examples of 'doubling' are illustrated in Figure 2.10.



(a) (b)

Figure 2.10 Two examples of doubling on an elliptic curve

Of course, the above examples are just two instances of addition. The general definition, which we are about to give, must account for more cases. We use the notation introduced in Lemma 2.10.

Definition 2.12 *Addition of points on an elliptic curve*

Let F be a field of characteristic distinct from 2 and 3, and let $E_F(a,b)$ be an elliptic curve over F. Let P, P_0, P_1 and P_2 be arbitrary points on $E_F(a,b)$ distinct from $\mathcal{O}$, with $P_1 \neq P_2$. We define a binary operation on $E_F(a,b)$ as follows:

(a) $\mathcal{O} + \mathcal{O} = \mathcal{O}$

(b) $P + \mathcal{O} = \mathcal{O} + P = P$

(c) $P + (-P) = \mathcal{O}$

(d) if P_0 has non-zero y-coordinate, then $2P_0 = P_0 + P_0 = -\pi'(P_0)$

(e) if P_1 and P_2 are such that $P_1 \neq \pm P_2$, then $P_1 + P_2 = -\pi(P_1, P_2)$.

Worked Exercise 2.13 *Computing $P_1 + P_2$ and $2P_0$*

Consider again the elliptic curve $E_{GF_{19}}(1,1)$ from Worked Exercise 2.11 and the three points $P_1 = (0,1)$, $P_2 = (5,6)$ and $P_0 = (9,-6)$ on the curve. Find the points $P_1 + P_2$ and $2P_0 = P_0 + P_0$.

Solution

By parts (e) and (d) of Definition 2.12 we have $P_1 + P_2 = -\pi(P_1, P_2)$ and $2P_0 = -\pi'(P_0)$. In Worked Exercise 2.11 we saw that $\pi(P_1, P_2) = (-4,-3)$ and $\pi'(P_0) = (7,3)$. It follows that $P_1 + P_2 = -(-4,-3) = (-4,3)$ and $2P_0 = -(7,3) = (7,-3)$.

The surprising feature of the operation of addition introduced in Definition 2.12 is that it makes an elliptic curve into an abelian group! The identity element of the group is our mysterious special point at infinity, $\mathcal{O}$, which explains its importance.

Proposition 2.14

Let F be a field of characteristic distinct from 2 and 3, let $E = E_F(a,b)$ be an elliptic curve over F, and let $+$ be the binary operation on E introduced in Definition 2.12. Then $(E,+)$ is an abelian group with identity element $\mathcal{O}$. The inverse of a point P is the point $-P$.

Proof We first address a fine point – namely, we need to check that the operation $+$ is well defined. By Definition 2.12, this amounts to checking that the points $\pi(P_1, P_2)$ and $\pi'(P_0)$ are uniquely defined. This is clear for $\pi(P_1, P_2)$, since this point is uniquely defined by part (a) of Lemma 2.10 whenever $P_1 \neq \pm P_2$.

The point $\pi'(P_0)$ is uniquely defined by part (b) of Lemma 2.10 whenever $P_0 = (x_0, y_0)$ is such that $y_0 \neq 0$. Note that if a point P has zero y-coordinate, then we have $P = -P$, and $P + P = \mathcal{O}$, by Definition 2.12. We conclude that the operation $+$ is indeed well defined.

Next we prove that $(E, +)$ is an abelian group. From the fact that $\pi(P_1, P_2) = \pi(P_2, P_1)$ and from Definition 2.12 it follows that the operation $+$ is commutative. Also, $\mathcal{O}$ is the identity element by Definition 2.12(b), and the inverse of a point $P \in E$ is $-P$ by Definition 2.12(c).

The only remaining issue is associativity of the operation $+$. However, the proof that this operation is associative either uses results that are beyond the scope of this module or requires a level of calculation where it is best to use a computer. This is why we omit this last part of the proof. ■

You may have been wondering about the reason for our assumption $4a^3 + 27b^2 \neq 0$ in Definition 2.3, where we introduced elliptic curves. By Lemma 2.1, for fields F of characteristic distinct from 2 and 3 the inequality $4a^3 + 27b^2 \neq 0$ implies that the polynomial $f(x) = x^3 + ax + b$ over F has no multiple roots. So, what would go wrong with our theory if we allowed polynomials $f(x) = x^3 + ax + b$ with a multiple root?

Suppose that $f(x) = x^3 + ax + b$ has a root x_0 of multiplicity 2 and a simple root $x_1 \neq x_0$, that is, $f(x) = (x - x_0)^2(x - x_1)$. Consider the points $P_1 = (x_1, y_1) = (x_1, 0)$ and $P_2 = (x_2, y_2) = (x_0, 0)$ on the corresponding elliptic curve $E_F(a, b)$. We show that in such a case the operation $+$ would not be associative (this is one of the numerous special cases one would have to go through in proving associativity in Proposition 2.14).

Indeed, when computing $\pi(P_1, P_2)$ by part (a) of Lemma 2.10 we would have $c = d = 0$, $y_3 = 0$ and $x_3 = -x_1 - x_2$, or, equivalently, $x_0 + x_1 + x_3 = 0$. Since x_3 is now a root of $f(x)$, the left-hand side of the last equation is the sum of the three roots of $f(x)$. Since we assumed x_0 to have multiplicity 2, $x_3 = x_0$. It follows that $-\pi(P_1, P_2) = -(x_0, 0) = (x_0, 0) = P_2$, and so by Definition 2.12 we would have $P_1 + P_2 = P_2$. Observe that by the same definition we have $P_2 + P_2 = \mathcal{O}$ since P_2 has zero y-coordinate. But then Definition 2.12 would give

$$(P_1 + P_2) + P_2 = P_2 + P_2 = \mathcal{O},$$

while

$$P_1 + (P_2 + P_2) = P_1 + \mathcal{O} = P_1,$$

which violates associativity.

Observe that a contradiction already follows from the equation $P_1 + P_2 = P_2$: if we assume that $(E, +)$ is a group, we can use the cancellation law and cancel P_2 on both sides, obtaining the contradictory statement $P_1 = \mathcal{O}$. But if we were to prove Proposition 2.14 systematically as indicated above, we would only discover the contradiction when checking associativity.

If $f(x) = x^3 + ax + b$ had a root x_0 of multiplicity 3, then we would have $x^3 + ax + b = (x - x_0)^3$. Comparing coefficients for x^2 would quickly give $x_0 = 0$, so that $f(x) = x^3$ with $a = b = 0$. In this case a similar contradiction as above can be obtained for the points $P_1 = (0, 0)$ and $P_2 = (1, 1)$.

This is a good place to clarify what happens to the points $P_1 = (x_1, 0)$, $P_2 = (x_2, 0)$ and $P_3 = (x_3, 0)$ if $f(x)$ has three distinct roots x_1, x_2 and x_3 in F. If we use Lemma 2.10(a) and argue as above, we find that $P_1 + P_2 = P_3$. Since $P_j = -P_j$ for $j \in \{1, 2, 3\}$, we also have $P_2 + P_3 = P_1$ and $P_3 + P_1 = P_2$. We see that, in this case, the set $\{\mathcal{O}, P_1, P_2, P_3\}$ forms a subgroup of $(E, +)$ isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$.

We conclude with a remark. From now on we will identify an elliptic curve $E = E_F(a, b)$ with the corresponding abelian group $(E, +)$. Also, as usual when working with abelian groups, if P is an element of an elliptic curve $(E, +)$, we will write $2P$ for $P + P$, $3P$ for $P + P + P$, and so on.

2.4 Examples of elliptic curves as abelian groups

This subsection is intended to provide practice of calculations in elliptic curves regarded as abelian groups. Remember that when we apply Lemma 2.10 to an elliptic curve over the finite field GF_p, all the calculations – in particular of c_0, d_0, c and d – are modulo p.

Worked Exercise 2.15 *The rational elliptic curve $E_{\mathbb{Q}}(1, 2)$*

Describe the group $(E_1, +)$ for the rational elliptic curve $E_1 = E_{\mathbb{Q}}(1, 2) = \{\mathcal{O}, (-1, 0), (1, \pm 2)\}$ from Example 2.6.

Solution

Since E_1 has four elements and there are only two groups of order 4, the group $(E_1, +)$ is either isomorphic to $\mathbb{Z}_4$ or to $\mathbb{Z}_2 \times \mathbb{Z}_2$ by Theorem 3.2 of Book B, Chapter 7. Let us try to compute the element $2P_0 = P_0 + P_0$ for $P_0 = (1, 2)$. Lemma 2.10(b) with $a = 1$, $b = 2$, $x_0 = 1$ and $y_0 = 2$ gives $c_0 = 1$, $d_0 = 1$, $x' = -1$ and $y' = 0$. It follows that $\pi'(P_0) = (x', y') = (-1, 0)$. Applying Definition 2.12 to $P_0 + P_0$ gives $2P_0 = -\pi'(P_0) = (-1, 0)$.

Since $2P_0 \neq \mathcal{O}$, the element P_0 does not have order 2 in the group $(E_1, +)$. In this rather special case, this finding already determines the group: as every non-identity element of $\mathbb{Z}_2 \times \mathbb{Z}_2$ has order 2, our group $(E_1, +)$ must be isomorphic to $\mathbb{Z}_4$. The reader is invited to check that $3P_0 = 2P_0 + P_0 = (1, -2)$, and so P_0 is a generator of $(E_1, +) \simeq \mathbb{Z}_4$.

Example 2.16 *The rational elliptic curve $E_{\mathbb{Q}}(1,1)$*

Another rational elliptic curve considered in Example 2.6 was $E_2 = E_{\mathbb{Q}}(1,1)$. We commented that this curve is infinite and, getting ahead of ourselves at that stage, we remarked that the group $(E_2, +)$ is isomorphic to the group of integers $\mathbb{Z}$.

Although you do not have the necessary mathematical tools to prove this claim, we can verify how some of the points of $E_{\mathbb{Q}}(1,1)$ listed in Example 2.6 are related in terms of the group operation. Let us begin with the point $P_0 = (x_0, y_0) = (0,1)$. First we determine $2P_0$. As in Worked Exercise 2.15, we apply Lemma 2.10(b) with $a = 1$, $b = 1$, $x_0 = 0$ and $y_0 = 1$, and we obtain $c_0 = \frac{1}{2}$, $d_0 = 1$, $x' = \frac{1}{4}$ and $y' = \frac{9}{8}$, so $\pi'(P_0) = (\frac{1}{4}, \frac{9}{8})$, and by Definition 2.12 we have $2P_0 = -\pi'(P_0) = (\frac{1}{4}, -\frac{9}{8})$.

Next, let us determine the sum of P_1 and P_2 for $P_1 = P_0 = (0,1) = (x_1, y_1)$ and $P_2 = 2P_0 = (\frac{1}{4}, -\frac{9}{8}) = (x_2, y_2)$. Lemma 2.10(a) gives $c = -\frac{17}{2}$, $d = 1$, $x_3 = 72$ and $y_3 = -611$, which implies that $\pi(P_1, P_2) = (72, -611)$. Definition 2.12 applied to $P_1 + P_2$ now tells us that $3P_0 = P_1 + P_2 = -\pi(P_1, P_2) = (72, 611)$.

By more advanced methods it can be shown that the point $P_0 = (0,1)$ is in fact a generator of $(E_2, +)$ and that this group is infinite (and hence isomorphic to $\mathbb{Z}$ by Theorem 2.1 of Book B, Chapter 6). Note also that the inverses of the elements calculated above, that is, $-P_0 = (0,-1)$, $-2P_0 = (\frac{1}{4}, \frac{9}{8})$ and $-3P_0 = (72, -611)$, are also in $(E_2, +)$, which agrees with the list of the first few elements of this group given in Example 2.6.

In the following exercises we calculate groups defined by elliptic curves over various small finite fields.

Worked Exercise 2.17 *An elliptic curve over GF_7*

Determine the group $(E_3, +)$, where E_3 is the elliptic curve $E_{GF_7}(0,2)$, corresponding to the equation $y^2 = x^3 + 2$ over GF_7.

Solution

Checking the seven possible values of x in GF_7 reveals that $x^3 + 2$ is a square over GF_7 if, and only if, $x \in \{0, -1, -2, 3\}$, which then gives the following list of elements of E_3:

$$E_3 = \{\mathcal{O}, (0, \pm 3), (-1, \pm 1), (-2, \pm 1), (3, \pm 1)\}.$$

We see that $|E_3| = 9$, and so by Theorem 3.2 of Book B, Chapter 7 there are only two candidates for the abelian group $(E_3, +)$, namely $\mathbb{Z}_9$ and $\mathbb{Z}_3 \times \mathbb{Z}_3$.

Take $P_0 = (0, 3)$. For the values $a = 0$, $b = 2$, $x_0 = 0$ and $y_0 = 3$, Lemma 2.10(b) returns $c_0 = 0$, $d_0 = \frac{4}{-1} = 3$, $x_0' = 0$ and $y' = 3$, hence $\pi'(P_0) = P_0$ and, by Definition 2.12, $2P_0 = -\pi'(P_0) = -P_0$. The last equality implies that $3P_0 = \mathcal{O}$, that is, the point $P_0 = (0, 3)$ has order 3 in the group $(E_3, +)$. By the same token, the point $-P_0 = (0, -3)$ has also order 3.

Now consider a different point P_0, say, $P_0 = (-1, 1)$. Lemma 2.10(b) applied to $a = 0$, $b = 2$, $x_0 = -1$ and $y_0 = 1$ gives $c_0 = \frac{3}{2} = -2$, $d_0 = \frac{5}{2} = -1$, $x' = -1$ and $y' = 1$. This gives $\pi'(P_0) = P_0$ again, which implies that $2P_0 = -P_0$ again. We have thus discovered another two points of order 3, namely, $(-1, 1)$ and its inverse, $(-1, -1)$.

Since the group $\mathbb{Z}_9$ contains just two elements of order 3, we conclude that the group $(E_3, +)$ is isomorphic to $\mathbb{Z}_3 \times \mathbb{Z}_3$.

As an aside, and addressing the remark immediately following the proof of Lemma 2.10, here we have met the case $\pi'(P_0) = P_0$. In fact, since every non-identity element in $(E_3, +)$ has order 3, it follows that $\pi'(P_0) = P_0$ for every non-identity element of E_3.

Exercise 2.4

Let $E_4 = E_{GF_5}(1, 1)$ be the elliptic curve given by the equation $y^2 = x^3 + x + 1$ over the field GF_5. Determine the group $(E_4, +)$.

Worked Exercise 2.18 *An elliptic curve over GF_{19}*

Determine the group $(E_5, +)$ for the elliptic curve $E_5 = E_{GF_{19}}(1, 4)$ over the field of order 19, corresponding to the equation $y^2 = x^3 + x + 4$ over GF_{19}.

Solution

In examples that involve a slightly larger group, like this one, it is of advantage to be organised. Since we will need to take square roots in $GF_{19} = \{0, \pm 1, \ldots, \pm 9\}$, we start by determining the set of non-zero squares in this field. This can be done in two ways:

- we find a primitive element of GF_{19} – then its even powers are the squares of GF_{19} – or
- we calculate the values of $j^2 \pmod{19}$ for $j \in \{1, \ldots, 9\}$; this range is sufficient since $j^2 = (-j)^2$.

By inspection, one finds that, for example, 2 is a primitive element of GF_{19}, so that the non-zero squares are 2^2, 2^4, ...

For the second approach, a table such as Table 2.3 below is useful.

Table 2.3 The non-zero squares of GF_{19}

j	1	2	3	4	5	6	7	8	9
$j^2 \pmod{19}$	1	4	9	−3	6	−2	−8	7	5

We then arrange the values of $f(x) = x^3 + x + 4$ for $x \in \{0, 1, \ldots, 9\}$ in another table, and in the cases when $f(x)$ is a square (which we may check quickly using Table 2.3) we also give the value of $y = \pm\sqrt{f(x)}$.

Table 2.4 The values of $f(x)$, $x \in \{0, 1, \ldots, 9\}$, with square roots

x	0	1	2	3	4	5	6	7	8	9
x^3	0	1	8	8	7	−8	7	1	−1	7
$f(x)$	4	6	−5	−4	−4	1	−2	−7	−8	1
y	±2	±5	×	×	×	±1	±6	×	±7	±1

The remaining values for $x \in \{-1, -2, \ldots, -9\}$ can be obtained from Table 2.4 by observing that $f(-x) = -f(x) + 8$, so that we may omit the row corresponding to x^3.

Table 2.5 The values of $f(x)$, $x \in \{-1, \ldots, -9\}$, with square roots

x	−1	−2	−3	−4	−5	−6	−7	−8	−9
$f(x)$	2	−6	−7	−7	7	−9	−4	−3	7
y	×	×	×	×	±8	×	×	±4	±8

A list of all pairs $(x, y) \in E_5 \backslash \{\mathcal{O}\}$ now follows from Tables 2.4 and 2.5:

Table 2.6 A list of all the elements of $E_5 \backslash \{\mathcal{O}\}$

x	0	1	5	−5	6	8	−8	9	−9
y	±2	±5	±1	±8	±6	±7	±4	±1	±8

It follows that $|E_5| = 19$. This is very fortunate, as it simplifies our task to determine $(E_5, +)$: since 19 is a prime and the only group of order 19 is the cyclic group $\mathbb{Z}_{19}$ (by Proposition 3.1 of Book B, Chapter 7), it follows that $(E_5, +) \simeq \mathbb{Z}_{19}$. In particular, any of the 18 points (x, y) listed in Table 2.6 is a generator of the group $(E_5, +)$.

Exercise 2.5

Determine the group $(E_6, +)$ for the elliptic curve $E_6 = E_{GF_{13}}(0, 2)$ over the field of order 13, corresponding to the equation $y^2 = x^3 + 2$ over GF_{13}.

Hint: you met this curve in Exercise 2.3.

Exercise 2.6

Determine the group $(E_7, +)$ for the elliptic curve $E_7 = E_{GF_{17}}(1, -1)$ over the field of order 17, corresponding to the equation $y^2 = x^3 + x - 1$ over GF_{17}.

Two substantial results on elliptic curves

The examples in this subsection suggest that determining the abelian group arising from an elliptic curve over the field of rational numbers or over a finite field is a hard problem. This is indeed the case. In what follows, we state just two main results in this direction, accompanied by a few comments.

The main result on groups arising from rational elliptic curves was a joint effort by L. Mordell (1922), A. Weil (1928) and B. Mazur (1977). In the statement of the result below, for any non-negative integer r the symbol $\mathbb{Z}^r$ denotes the trivial group if $r = 0$, and the direct product

$$\underbrace{\mathbb{Z} \times \cdots \times \mathbb{Z}}_{r \text{ times}} \quad \text{if} \quad r \geq 1.$$

Theorem 2.19

Let E be an elliptic curve over the field of rational numbers. Then the group $(E, +)$ is isomorphic to a group of the form $A \times \mathbb{Z}^r$ for some $r \geq 0$, where A is a finite abelian group isomorphic to $\mathbb{Z}_n$ for some $n \in \{1, 2, \ldots, 9, 10, 12\}$, or to $\mathbb{Z}_2 \times \mathbb{Z}_{2m}$ for some $m \in \{1, 2, 3, 4\}$.

It is interesting to note that the relationship of the parameter r, called the *rank* of E, to elliptic curves arising from the rational curve E by 'reduction modulo p' for all sufficiently large prime numbers p, is the subject of the famous *Birch and Swinnerton–Dyer Conjecture*. The conjecture, too advanced to be formulated here, is one of the seven Millennium Problems listed by the Clay Mathematics Institute in 2000, each carrying a prize of 1 000 000 USD for the first correct proof.

A sharp bound on the order of the abelian groups arising from elliptic curves over finite fields was proved by H. Hasse (1933). We will state the bound in combination with a classical result about the structure of these groups.

Theorem 2.20

Let E be an elliptic curve over a finite field of order q. Then the group $(E, +)$ is either cyclic, or isomorphic to a product $\mathbb{Z}_k \times \mathbb{Z}_m$ for some $k, m \geq 2$ such that k is a divisor of m. In either case, the order e_q of the group $(E, +)$ satisfies the inequality

$$|e_q - (q+1)| \leq 2\sqrt{q}.$$

You may want to check that the orders of the groups determined in Example 2.7, Exercise 2.4 and Worked Exercises 2.17 and 2.18 satisfy the structure statement and the inequality in Theorem 2.20.

2.5 Two cryptosystems based on elliptic curves

We begin with the same motivational example as in our discussion of the cryptosystems based on modular arithmetic: imagine that Alice wants to send a message to Bob in such a way that Eve, the eavesdropper, has no realistic chance to know the message, even though Eve knows all the details of the encryption process (except, of course, the message itself). This time, however, we want to employ elliptic curves to do the job.

Among a variety of existing cryptosystems based on elliptic curves, we will present two: the Diffie–Hellman–ElGamal and the Menezes–Vanstone systems. Later in this subsection we will briefly discuss the security of these two systems and some related computational issues.

The Diffie–Hellman–ElGamal elliptic curve cryptosystem

The Diffie–Hellman–ElGamal elliptic curve cryptosystem is based on the 'shared secret' idea of the Diffie–Hellman cryptosystem which you encountered in Subsection 1.5, combined with ElGamal encryption and decryption on an elliptic curve over a finite field. Suppose, as usual, that Alice and Bob want to communicate securely through an insecure channel. In order to set up the communication, Alice and Bob choose an elliptic curve over a finite field together with some parameters, rather like in the setup of a Diffie–Hellman exchange scheme.

Communication setup

Alice and Bob agree a prime p and two elements $a, b \in GF_p$ such that $4a^3 + 27b^2 \neq 0$, so that the selected parameters define an elliptic curve $E = E_{GF_p}(a, b)$. Recall that for a fast generation of large primes with high probability they can use the Rabin–Miller test described in Subsection 1.4. They also agree an initial point $P \in E$; a quick way of generating such a point will be discussed later.

Alice then chooses a positive integer s_A such that $1 < s_A < |E| - 1$. At the other end of the communication line, Bob also chooses a positive integer s_B such that $1 < s_B < |E| - 1$. The integers s_A and s_B are kept secret. Alice then calculates the point $P_A = s_A P \in E$, that is, modulo p, and sends P_A over the channel to Bob. Symmetrically, Bob calculates the point $P_B = s_B P \in E$ and sends P_B to Alice. In this way, both P_A and P_B have been made public.

As part of the setup, Alice also computes the point $s_A P_B$ in E using her private key s_A, and Bob computes the point $s_B P_A$ with the help of his private key s_B. Since $s_A P_B = s_A(s_B P) = s_B(s_A P) = s_B P_A$, Alice and Bob obtain the same point. We will denote this resulting point by Q; this will not be released and will remain a shared secret between Alice and Bob. The shared secret Q will not play any role in what follows, but is important in practice; we will briefly touch upon this issue at the end of this chapter.

To sum up, the quadruple (E, P, P_A, P_B) is the public key in this system, and the integers s_A and s_B are respectively Alice's and Bob's private keys.

Encryption

Suppose Alice wants to send Bob a message. First she must convert her message into a point on the elliptic curve E; a way to do this will be introduced later. For now, assume that the message Alice wants to send has been encoded in the form of a point M on E. She generates a random positive integer k, $1 < k < |E|$, and she uses the public key (E, P, P_A, P_B) to compute two points in the group $(E, +)$: the point $M_1 = kP$ and the point $M_2 = M + kP_B$. Note that only M_2 depends on the message M; both M_1 and M_2, however, depend on the integer k known only to Alice. She then sends M_1 and M_2 to Bob through the communication channel, so that anybody, including Eve, can get hold of M_1 and M_2.

Decryption

After receiving M_1 and M_2, Bob performs an amazingly simple calculation: using his private key, s_B, he computes the point $M_2 - s_B M_1$ in the group $(E, +)$. But even without knowledge of k, the way M_1 and M_2 have been created and the fact that $P_B = s_B P$ give

$$M_2 - s_B M_1 = (M + kP_B) - s_B(kP) = M + k(s_B P) - s_B(kP) = M$$

and hence $M_2 - s_B M_1$ is equal to the original message M (as a point on the curve E).

Comments

In this setting, communication can go both ways, as in the Diffie–Hellman system. If Bob wants to send Alice a message $M' \in E$, he generates a random integer k', $1 < k' < |E|$, and computes $M'_1 = k'P$ and $M'_2 = M' + k'P_A$. After receiving M'_1 and M'_2, Alice decrypts Bob's message by computing $M'_2 - s_A M'_1$, which is equal to M' by an analogous computation as above.

A deeper analysis, details of which are beyond the scope of this module, shows that if Eve had a general method to decipher Alice's message M and Bob's message M', then she would be able to determine their secret keys, too. To make this cryptosystem as secure as possible, one has to make sure that attempts to determine s_A and s_B from the knowledge of the points P, $P_A = s_A P$ and $P_B = s_B P$ are hopeless. We will discuss the security of this cryptosystem later in this chapter. For now we just note what you would expect – namely, the prime p we have begun with should be large to make breaking attempts infeasible.

In our description, we have assumed that Alice's message is a point of an elliptic curve $E = E_{GF_p}(a, b)$. In other words, we have assumed that M is of the form $M = (x, y)$, where x and y are elements of GF_p that satisfy the equation $y^2 = x^3 + ax + b$ modulo p. Now, if p is extremely large, how can Alice and Bob find solutions of this equation at all? And, assuming they have a procedure for finding the solutions, how can they convert a plain text message into a point on the curve?

Before addressing these questions, we give a concrete example of encryption and decryption with the help of an elliptic curve.

Example 2.21 *The Diffie–Hellman–ElGamal system: encryption and decryption*

Suppose that Alice and Bob have chosen $p = 19$, $a = 1$ and $b = 4$. This means that their elliptic curve is $E = E_{GF_{19}}(1, 4)$, corresponding to the equation $y^2 = x^3 + x + 4$ (in GF_{19}, that is, modulo 19). Note that we worked with this curve in Worked Exercise 2.18. Suppose further that they have agreed to use the point $P = (0, 2) \in E$.

Bob decides to have $s_B = 4$ as his private key and needs to compute the point $P_B = s_B P$. You may check that an evaluation using Definition 2.12(d) and Lemma 2.10(b) gives $2P = (6, 6)$ and $4P = 2(2P) = (-8, 4)$. It follows that Bob's point is $P_B = (-8, 4)$, which he sends through the channel.

Suppose Alice wants to send Bob the message $M = (1, 5) \in E$. According to the procedure outlined above, Alice randomly chooses a positive integer, say $k = 3$ for the sake of this example. She now needs to determine $M_1 = 3P$ and $M_2 = M + 3P_B$. To determine $3P$ she first determines $2P$ and then $3P = P + 2P$. Since we know that $2P = (6, 6)$, Definition 2.12(e) and Lemma 2.10(a) imply that $M_1 = 3P = P + 2P = (5, 1)$. Similarly, from $P_B = (-8, 4)$ she obtains $2P_B = (1, -5)$ and $3P_B = P_B + 2P_B = (8, -7)$. Finally, Alice computes $M_2 = M + 3P_B = (-8, 4)$ and sends $M_1 = (5, 1)$ and $M_2 = (-8, 4)$ to Bob through the communication channel.

After receiving M_1 and M_2, Bob calculates $s_B M_1 = 4M_1$; we omit the details of Bob's calculations and just note that he obtains $4M_1 = (8, -7)$. He can then recover the message by calculating $M_2 - 4M_1$; we look at this last calculation in somewhat more detail. We have

$$M_2 + (-4M_1) = (-8, 4) + (8, 7).$$

Lemma 2.10(a) applied to the values $a = 1$, $b = 4$, $(x_1, y_1) = (-8, 4)$ and $(x_2, y_2) = (8, 7)$ gives, modulo 19,

$$c = \frac{7-4}{8-(-8)} = \frac{3}{-3} = -1,$$
$$d = \frac{32+56}{-3} = \frac{-7}{-3} = -4,$$
$$x_3 = 1 - 8 + 8 = 1, \quad \text{and finally}$$
$$y_3 = -1 - 4 = -5.$$

By Definition 2.12(e), we conclude that $M_2 - 4M_1 = (1, 5)$, which is indeed the original message.

Exercise 2.7

Alice and Bob agree to use the Diffie–Hellman–ElGamal system with the elliptic curve $E = E_{GF_{17}}(1, -1)$ from Exercise 2.6. They use the public key (E, P, P_A, P_B), where $P = (0, 4)$, $P_A = (4, 4)$, $P_B = (-4, -4)$. Alice's and Bob's private keys are $s_A = 2$ and $s_B = 3$ respectively. Alice wants to send Bob the message $M = (8, -3)$ using the randomly generated integer $k = 5$. Determine Alice's ciphertext and verify its correct decryption by Bob.

The Menezes–Vanstone elliptic curve cryptosystem

If Alice and Bob were to use the Diffie–Hellman–ElGamal system to exchange a large number of messages, they would have to have a quick means to generate points on an elliptic curve to encode their messages. While, as we shall see later, this is not an obstacle in reality, there is a way to bypass this requirement and encrypt messages in a slightly different form that avoids repeated generation of points on an elliptic curve. The method was proposed by Menezes and Vanstone in 1993.

Communication setup

This step is identical to the setup of Diffie–Hellman–ElGamal cryptosystem: the public key is a quadruple (E, P, P_A, P_B), and Alice's and Bob's private key are s_A and s_B respectively, as before.

Encryption

Alice's message now has the form of an ordered pair (m_1, m_2), where $m_1, m_2 \in \mathbb{Z}_p$ and p is the order of the field GF_p over which Alice's and Bob's elliptic curve is defined. So now Alice does not need to generate points on E. Instead, she chooses a random positive k such that $1 < k < |E|$. Knowing P and P_B allows her to compute the points $kP \in E$ and $kP_B \in E$. Let the coordinates of the second point be $kP_B = (d_1, d_2)$. Alice finally computes $c_1 = d_1 m_1 \in \mathbb{Z}_p$ and $c_2 = d_2 m_2 \in \mathbb{Z}_p$. Her ciphertext is the triple (kP, c_1, c_2), which she sends to Bob through the communication channel.

Decryption

To decrypt this ciphertext, Bob first determines the pair (d_1, d_2) by multiplying the point kP in Alice's message by his private key s_B, since

$$s_B(kP) = k(s_BP) = kP_B = (d_1, d_2).$$

Using the Euclidean Algorithm as in the solution to Worked Exercise 1.2, he then determines d_1^{-1} and d_2^{-1} modulo p. Finally, Bob recovers Alice's message (m_1, m_2) by multiplying the second and third components of her ciphertext, c_1 and c_2, by d_1^{-1} and d_2^{-1} modulo p. Indeed, $m_1 = c_1 d_1^{-1}$ and $m_2 = c_2 d_2^{-1}$, all modulo p.

Comments

The symmetry of this scheme allows Alice and Bob to communicate both ways. This scheme greatly enlarges the number of messages Alice can send, as her messages are now ordered pairs of elements of $\mathbb{Z}_p$, rather than points on an elliptic curve over GF_p. The number of such ordered pairs is p^2, which is much larger than the number of points on an elliptic curve over GF_p, which is approximately p by Hasse's Theorem (Theorem 2.20).

Example 2.22 *The Menezes–Vanstone system: encryption and decryption*

We will again work with the elliptic curve $E = E_{GF_{19}}(1, 4)$, corresponding to the equation $y^2 = x^3 + x + 4$ in GF_{19}, as in Example 2.21. To simplify matters, let us assume that Alice and Bob agree to use the same base point $P = (0, 2)$ as in Example 2.21. Suppose further that Alice and Bob choose the values $s_A = 4$ and $s_B = 3$ as their private keys, respectively.

We will now use the results of our computations in Example 2.21. Alice computes $s_AP = 4P = (-8, 4)$, Bob determines $s_BP = 3P = (5, 1)$, and they make these results public by sending them to each other through the communication channel.

As in the examples illustrating the public-key cryptosystems in Section 1, suppose that Alice wants to send Bob the three-digit security number, 609, on yet another credit card. Assume that Alice and Bob have agreed to form a four-digit number out of the security number by inserting a zero in the front, so that Alice's message is $(m_1, m_2) = (06, 09)$.

Suppose that Alice has chosen $k = 5$. With the help of Definition 2.12 and Lemma 2.10, she computes $kP = 5P = (-5, -8)$ and $k(P_B) = 5(5, 1) = (-8, -4)$. Thus, Alice finds that $d_1 = -8$ and $d_2 = -4$. This enables her to compute

$$c_1 = d_1 m_1 = -48 \equiv 9 \pmod{19} \quad \text{and}$$
$$c_2 = d_2 m_2 = -36 \equiv 2 \pmod{19}.$$

Alice then sends the ciphertext $(kP, c_1, c_2) = ((-5, -8), 9, 2)$ through an insecure channel to Bob.

To decrypt the ciphertext, Bob computes $s_B(kP) = 3(-5,-8) = (-8,-4)$, so that $d_1 = -8$ and $d_2 = -4$. Inverting d_1 and d_2 he obtains $d_1^{-1} = 7$ and $d_2^{-1} = -5$. He finally recovers

$$m_1 = c_1 d_1^{-1} = \quad 63 \equiv 6 \pmod{19} \quad \text{and}$$
$$m_2 = c_2 d_2^{-1} = -10 \equiv 9 \pmod{19}.$$

Bob can now concatenate $m_1 = 06$ and $m_2 = 09$, and by deleting the leftmost zero he obtains Alice's security number, 609.

Exercise 2.8

Alice and Bob choose to use the Menezes–Vanstone cryptosystem with the elliptic curve $E = E_{GF_{17}}(1,-1)$ from Exercise 2.7. They use again the public key (E, P, P_A, P_B), where

$$P = (0,4),\ P_A = (4,4),\ P_B = (-4,-4),$$

and their private keys are $s_A = 2$ and $s_B = 3$ respectively. Alice wants to send Bob the message $(m_1, m_2) = (2,4)$ using the randomly generated integer $k = 4$. Determine Alice's ciphertext and verify its correct decryption by Bob.

Security of cryptosystems based on elliptic curves

In general, it would seem that the only feasible way to break the cryptosystems based on elliptic curves that we have just described is to determine Alice's and Bob's private keys, s_A and s_B, from the equations $P_A = s_A P$ and $P_B = s_B P$, where the three points P, P_A and P_B lying on an elliptic curve are public knowledge. This is a special case of a more general problem, the *discrete logarithm problem for elliptic curves*. The discrete logarithm for elliptic curves is defined next.

Definition 2.23 *Discrete logarithm for elliptic curves*

Let $E = E_{GF_p}(a,b)$ be an elliptic curve over a Galois field GF_p. Let $P, P' \neq \mathcal{O}$ be two points on E, regarded as elements of the abelian group $(E,+)$. If there exists an integer s, $1 < s < |E|$, such that $P' = sP$, then s is called the **discrete logarithm of P' to the base P**.

Observe the analogy with Definition 1.15 of the discrete logarithm for finite fields. The difference is that in the context of elliptic curves the exponentiation operation is replaced by multiplication. Moreover, here, if $(E,+)$ is a cyclic group and P is a generator of this group, then the discrete logarithm to the base P exists for any point $P \in E$.

It turns out that there are no known efficient algorithms to solve the discrete logarithm problem for elliptic curves, that is, to work out the

discrete logarithm of an elliptic curve to a given base. The best known algorithms to break a cryptosystem based on an elliptic curve defined over a field GF_p need about $c \times \sqrt{p}$ operations, where c is a constant. This vastly exceeds, for example, the number of operations needed to factorise a number of a similar size by the general number sieve algorithm. So in the case of elliptic curves we are in a better position, since elliptic curve cryptosystems defined over GF_p appear to be safe when p is a prime number about 250 bits long. In fact, elliptic curve cryptosystems appear to prevail and outperform other systems.

Fast generation of points on an elliptic curve

The last question to be addressed for elliptic curve cryptosystems is how one can generate points on an elliptic curve quickly, that is, in real time.

Let $E = E_{GF_p}(a, b)$ be an elliptic curve over the finite field GF_p, where p is a large odd prime. We know that $E\backslash\{\mathcal{O}\}$ is the set of all ordered pairs (x, y) such that $y^2 = x^3 + ax + b$. The task is to generate a point $P = (x, y)$ on E, that is, to find a solution (x, y) of the equation $y^2 = x^3 + ax + b$. By Hasse's Theorem we know that there are at least $p - 2\sqrt{p}$ such points, but the theorem does not tell us how to find them. We would like to emphasise that points of the form $P = (x, 0)$ are not of much use, since by Definition 2.12 doubling gives $2P = \mathcal{O}$.

There is an astonishingly simple method for generating such points quickly. As in the case of generating prime numbers in Subsection 1.4, it is based on probabilistic considerations. Certain results in probability theory tell us that if x is a random non-zero element of GF_p, the element $x^3 + ax + b$ may also be considered to be random (and non-zero, by the above remark). For such an x to generate a point (x, y) on E, the (random) element $x^3 + ax + b$ would have to have a square root y. This is the same as saying that $x^3 + ax + b$ would have to be a square, or, in the terminology of Definition 1.2 of Book C, Chapter 10, a quadratic residue. Now, what are the chances that a random element of GF_p is a square? Since for an odd P the field GF_p contains the same number of non-zero squares and non-squares, that is, $(p - 1)/2$, the probability of a randomly chosen element in GF_p to be a square is $\frac{1}{2}$.

This suggests that points on E can be found as follows. Suppose for simplicity that $p \equiv -1 \pmod 4$ and let $f(x) = x^3 + ax + b$. Take a random $x \in GF_p$ and successively test the values $f(x)$, $f(x + 1)$, $\ldots$, $f(x + j)$ by Euler's Criterion for quadratic residues (Subsection 1.1, Lemma 1.9). As indicated above, the probability that none of these are a quadratic residue modulo p is about 2^{-j}, which gets extremely small as j gets larger. So, with high probability, within a few steps this method generates an x such that $f(x)$ is a square modulo p. If, in addition, $p \equiv -1 \pmod 4$, one can quickly determine the solutions $\pm y$ of $y^2 = f(x)$ using the second part of Lemma 1.9.

This procedure can also be adapted to encode plaintext messages in the form of points on E. Indeed, as we know, any plaintext messages can be identified with a decimal number; we assume that the number is a positive integer z such that $z < \frac{|E|}{1000}$, say. Now, among the numbers $f(100z)$, $f(100z+1)$, $\ldots$, $f(100z+99)$ with high probability there will be one, say $f(100z+j)$ for some $j \in \{0,1,\ldots,99\}$, such that $f(100z+j)$ is a square in GF_p. Let $x = 100z + j$. Then the point $M = (x, y)$ with $y^2 = f(x)$ can then be taken to be the point on E encoding the message M. The person decrypting the message then simply discards the last two decimal digits of x to obtain z.

We reiterate that all these computations can be done extremely quickly by the squaring method introduced in Subsection 1.1.

A note on practical aspects

The ideas explained in this chapter have been automated and integrated into communication systems.

In a very simplified model, suppose that Alice and Bob want to communicate securely over the internet. When, say, Alice sends a request to Bob, her browser and Bob's browser first 'shake hands' and select a cryptosystem to be used for their communication. The selection is made from a rather narrow predefined set of cryptosystems that meet high international standards.

The most recommended model of secure communication appears to be a Diffie–Hellman key exchange combined with other techniques that require the shared secret as a seed for further encryption. The details, however, are way too complex to be included in this brief overview of algebraic methods used in public-key cryptography.

Learning outcomes

We list those topics on which we may set assessment questions to test your understanding of the chapter.

After working through this chapter you should be able to:

(a) Find the highest common factor of two appropriate integers using the Fast Euclidean Algorithm.

(b) Where appropriate find the multiplicative inverse of one integer modulo another by reversing the Fast Euclidean Algorithm.

(c) Where appropriate find a witness of non-primality.

(d) Find a primitive element of GF_p provided p is not too large.

(e) Check whether integers are quadratic residues mod p and find square roots in GF_p provided p is not too large.

(f) Use the RSA cryptosystem for suitable primes.

(g) Use the Diffie–Hellman cryptosystem for a suitable prime.

(h) Identify points on an elliptic curve in appropriate cases.

(i) Draw an elliptic curve in simple cases.

(j) Compute $\pi(P_1, P_2)$ and $\pi\prime(P_0)$ for given points P_0, P_1, P_2 on an elliptic curve.

(k) Identify the group $(E, +)$ for an elliptic curve in very simple cases.

(l) Use the Diffie–Hellman–ElGamal cryptosystem in simple cases.

(m) Use the Menezes–Vanstone cryptosystem in simple cases.

Solutions to exercises

Solution to Exercise 1.1

Suppose that $\text{hcf}(r, s) = h$. Then h is a common factor of r and $r - s$ by Theorem 4.4(e) of Book A, Chapter 1.

Now let k be a common factor of r and $r - s$. Then $km = r$ and $kn = r - s$ for some integers m and n. But then

$$kn = r - s = km - s,$$

which implies that $s = k(m - n)$. Hence k is a common factor of r and s. It follows that k divides h, so h is the highest common factor of r and $r - s$.

Solution to Exercise 1.2

Since $19\,191 < \frac{91\,919}{2}$, the input for step S_0 in the Fast Euclidean Algorithm is the pair $(91\,919, 19\,191)$. In the notation of the solution to Worked Exercise 1.1, we obtain the following steps.

$$\begin{array}{lrcll}
S_0: & 91\,919 & = & 4 \times \underline{19\,191} + \underline{\underline{15\,155}} & \text{use } \underline{4036} = 19\,191 - 15\,155 \text{ in } S_1 \\
S_1: & 19\,191 & = & 4 \times \underline{4036} + \underline{\underline{3047}} & \text{use } \underline{989} = 4036 - 3047 \text{ in } S_2 \\
S_2: & 4036 & = & 4 \times \underline{989} + \underline{\underline{80}} & \text{use } \underline{80} \text{ in } S_3 \\
S_3: & 989 & = & 12 \times \underline{80} + \underline{\underline{29}} & \text{use } \underline{29} \text{ in } S_4 \\
S_4: & 80 & = & 2 \times \underline{29} + \underline{\underline{22}} & \text{use } \underline{7} = 29 - 22 \text{ in } S_5 \\
S_5: & 29 & = & 4 \times \underline{7} + \underline{\underline{1}} &
\end{array}$$

Therefore $\text{hcf}(91\,919, 19\,191) = 1$.

Solution to Exercise 1.3

The multiplicative inverse of 19 191 modulo 91 919 exists by Exercise 1.2. Taking the equation in step S_5 of this exercise, working backwards and taking into account the replacements in the Fast Euclidean Algorithm (as in the solution to Worked Exercise 1.2) gives

$$\begin{aligned}
1 &= 29 - 7 \times 4 = 29 - \underline{(29 - 22)} \times 4 \\
&= 29 \times (-3) + 22 \times 4 \\
&= 29 \times (-3) + (80 - 2 \times 29) \times 4 \\
&= 80 \times 4 - 29 \times 11 \\
&= 80 \times 4 - (989 - 12 \times 80) \times 11 \\
&= 989 \times (-11) + 80 \times 136 \\
&= 989 \times (-11) + (4036 - 4 \times 989) \times 136 \\
&= 4036 \times 136 - 989 \times 555 = 4036 \times 136 - \underline{(4036 - 3047)} \times 555 \\
&= 4036 \times (-419) + 3047 \times 555 \\
&= 4036 \times (-419) + (19\,191 - 4 \times 4036) \times 555 \\
&= 19\,191 \times 555 - 4036 \times 2639 = 19\,191 \times 555 - \underline{(19\,191 - 15\,155)} \times 2639 \\
&= 19\,191 \times (-2084) + 15\,155 \times 2639 \\
&= 19\,191 \times (-2084) + (91\,919 - 4 \times 19\,191) \times 2639 \\
&= 91\,919 \times 2639 - 19\,191 \times 12\,640.
\end{aligned}$$

The multiplicative inverse of 19 191 modulo 91 919 is $-12\,640$, which modulo 91 919 is the same as $91\,919 - 12\,640 = 79\,279$.

Solution to Exercise 1.4

Observe that $899 = 29 \times 31$, $840 = (29-1) \times (31-1)$ and $841 = 29^2$. Let $n = 899, p = 29$, $q = 31$ and $d = e = 29$. Then $n = pq = 899$, $\phi(n) = 840$ and $de \equiv 1 \pmod{\phi(n)}$. By Lemma 1.3 we have $x^{841} \equiv x \pmod{899}$ for all integers x.

Solution to Exercise 1.5

We test the simplest b, that is, $b = 2$. Since $k - 1 = 288 = 2^5 \times 9$, we have $r = 5$ and $s = 9$. We evaluate the following congruences modulo 289 by repeatedly raising the left-hand side to the power 2.

$$\begin{aligned} 2^{9} &\equiv -66 \\ 2^{2\times 9} &\equiv 21 \\ 2^{2^2\times 9} &\equiv -137 \\ 2^{2^3\times 9} &\equiv -16 \\ 2^{2^4\times 9} &\equiv -33 \end{aligned}$$

By Corollary 1.5 the integer $b = 2$ is a witness of non-primality of $k = 289$ (note that $289 = 17^2$).

Solution to Exercise 1.6

Since $p = 257$ and $p - 1 = 256 = 2^8$, Lemma 1.7 tells us that a non-zero $\xi \in GF_{257}$ is a primitive element if, and only if, $\xi^{2^7} \not\equiv 1 \pmod{257}$. The element $\xi = 2$ will not work since $2^{2^3} \equiv -1 \pmod{257}$ and so $2^{2^7} \equiv 1 \pmod{257}$. Let us try $\xi = 3$. By squaring we find that, modulo 257,

$$\begin{aligned} 3^{2^2} &\equiv 81 \\ 3^{2^3} &\equiv 136 \\ 3^{2^4} &\equiv -8 \\ 3^{2^5} &\equiv 64 \\ 3^{2^6} &\equiv -16 \\ 3^{2^7} &\equiv -1. \end{aligned}$$

We conclude that $\xi = 3$ is a primitive element of GF_{257} by Lemma 1.7.

Solution to Exercise 1.7

Here $p = 193$ and $\frac{p-1}{2} = 96$. We have $(-1)^{96} \equiv 1 \pmod{193}$, and with the help of the calculations in the solution of Worked Exercise 1.8 we obtain $3^{96} = (3^8)^{12} = (-1)^{12} \equiv 1 \pmod{193}$ and $(-5)^{96} = 5^{96} \equiv -1 \pmod{193}$. By Lemma 1.9, we see that -1 and 3 are squares modulo 193 while -5 is not. Note that 5 is a primitive element of GF_{193} and hence it cannot be a square; a product of a square and a non-square is a non-square, which gives another proof that -5 is a non-square.

As $193 \equiv 1 \pmod{4}$, we cannot use the second part of Lemma 1.9 to determine the square roots of -1 and 3. With relatively small moduli, experiments with squares of integers usually help. Since $14^2 = 196 \equiv 3 \pmod{193}$, the square roots of $3 \pmod{193}$ are ± 14. Further, observe that $31^2 \equiv -4 \pmod{193}$; multiplying this congruence by 2^{-1} gives $(31 \times 2^{-1})^2 \equiv -1 \pmod{193}$. The multiplicative inverse 2^{-1} of 2 modulo 193 is $2^{-1} = 97$, and since $31 \times 97 \equiv 112 \pmod{193}$ we conclude that the square roots of -1 modulo 193 are ± 112.

Solution to Exercise 1.8

(a) Using the method and notation in the solution to Example 1.12, Alice obtains the following congruences modulo 5963.

$$
\begin{aligned}
&m_0 = 1111,\\
&m_0^2 = 1\,234\,321, \quad && m_1 \equiv m_0^2 \equiv m^{2^1} \equiv 5943\\
&m_1^2 = 35\,319\,249, && m_2 \equiv m_1^2 \equiv m^{2^2} \equiv 400\\
&m_2^2 = 160\,000, && m_3 \equiv m_2^2 \equiv m^{2^3} \equiv 4962\\
&m_3^2 = 24\,621\,444, && m_4 \equiv m_3^2 \equiv m^{2^4} \equiv 217\\
&m_4^2 = 47\,089, && m_5 \equiv m_4^2 \equiv m^{2^5} \equiv 5348\\
&m_5^2 = 28\,601\,104, && m_6 \equiv m_5^2 \equiv m^{2^6} \equiv 2556\\
&m_6^2 = 6\,533\,136, && m_7 \equiv m_6^2 \equiv m^{2^7} \equiv 3651
\end{aligned}
$$

Alice can then determine the product $m^{203} \equiv m_7 m_6 m_3 m_1 m_0 \pmod{5963}$:

$m_7 m_6 \equiv 5824 \pmod{5963}$

$m_7 m_6 m_3 \equiv 5824 m_3 \equiv 1990 \pmod{5963}$

$m_7 m_6 m_3 m_1 \equiv 1990 m_1 \equiv 1941 \pmod{5963}$.

Finally, she obtains her ciphertext:

$$c \equiv m_7 m_6 m_3 m_1 m_0 \equiv 1941 m_0 = 2\,156\,451 \equiv 3808 \pmod{5963}.$$

(b) Bob begins the decryption procedure by writing his private key as a sum of powers of two:

$$d = 515 = 2^9 + 2^1 + 2^0.$$

Letting $c_0 = c$, he successively computes $c_1 \equiv c^{2^1} \pmod{n}$, $c_2 \equiv c_1^2 = c^{2^2} \pmod{n}$, $c_3 \equiv c_2^2 = c^{2^3} \pmod{n}$, and so on, up to $c_9 \equiv c_8^2 = c^{2^9} \pmod{n}$, and using these he finally determines

$$c^{515} = c^{2^9 + 2^1 + 2^0} \equiv c_9 c_1 c_0 \pmod{n}.$$

These individual steps give the following congruences modulo 5963:

$$
\begin{aligned}
&c_0 = 3808,\\
&c_0^2 = 14\,500\,864, \quad && c_1 \equiv c_0^2 \equiv c^{2^1} \equiv 4811\\
&c_1^2 = 23\,145\,721, && c_2 \equiv c_1^2 \equiv c^{2^2} \equiv 3318\\
&c_2^2 = 11\,009\,124, && c_3 \equiv c_2^2 \equiv c^{2^3} \equiv 1426\\
&c_3^2 = 2\,033\,476, && c_4 \equiv c_3^2 \equiv c^{2^4} \equiv 93\\
&c_4^2 = 8649, && c_5 \equiv c_4^2 \equiv c^{2^5} \equiv 2686\\
&c_5^2 = 7\,214\,596, && c_6 \equiv c_5^2 \equiv c^{2^6} \equiv 5329\\
&c_6^2 = 28\,398\,241, && c_7 \equiv c_6^2 \equiv c^{2^7} \equiv 2435\\
&c_7^2 = 5\,929\,225, && c_8 \equiv c_7^2 \equiv c^{2^8} \equiv 2003\\
&c_8^2 = 4\,012\,009, && c_9 \equiv c_8^2 \equiv c^{2^9} \equiv 4873.
\end{aligned}
$$

Bob can now calculate $c^{227} \equiv c_9 c_1 c_0 \pmod{5963}$:

$$c_9 c_1 \equiv 3450.$$

The final result Bob obtains is

$$c^{515} \equiv c_9 c_1 c_0 \equiv 3450 c_0 \equiv 1111,$$

which is indeed Alice's intended message.

For completeness, note that $n = 5963 = pq$, where $p = 67$, $q = 89$, and $\phi(n) = (p-1)(q-1) = 5808$; one may verify that $e = 203$ and $d = 515$ satisfy $de \equiv 1 \bmod \phi(n)$.

Solution to Exercise 1.9

In this solution, calculations are all modulo 193.

Alice needs to determine $\alpha = \xi^a = 5^{12}$ and Bob needs to determine $\beta = \xi^b = 5^{17}$. Noting that $12 = 2^3 + 2^2$ and $17 = 2^4 + 2^0$ and using $5^{2^0} = 5$, $5^{2^1} \equiv 25$, $5^{2^2} \equiv 46$, $5^{2^3} \equiv 186$ and $5^{2^4} \equiv 49$, Alice obtains $\alpha \equiv 186 \times 46 \equiv 64$ and Bob obtains $\beta \equiv 49 \times 5 \equiv 52$.

One may verify that $\beta^a \equiv \alpha^b \equiv 64 \pmod{193}$, giving the value $\sigma = 64$. The multiplicative inverse σ^{-1} can be determined by the Euclidean Algorithm: since $193 = 3 \times 64 + 1$, it follows immediately that $\sigma^{-1} \equiv -3 \equiv 190 \pmod{193}$.

Alice encrypts the message $m = 123$ by evaluating $c = m\sigma = 123 \times 64 \equiv 152 \pmod{193}$.

Bob decrypts the ciphertext by computing $c\sigma^{-1} = 152 \times (-3) \equiv 123 \pmod{193}$.

Solution to Exercise 2.1

We prove the contrapositive, so we assume that $f(x)$ has at least one multiple root. Suppose that x_0 is a root of $f(x)$ of multiplicity at least two and let x_1 be the third root of $f(x)$. Then we have

$$f(x) = (x - x_0)^2(x - x_1) = x^3 - (2x_0 + x_1)x^2 + (x_0^2 + 2x_0x_1)x - x_0^2x_1.$$

But we have also assumed that $f(x) = x^3 + ax + b$. Comparing the coefficients of x^2 gives $2x_0 + x_1 = 0$, that is, $x_1 = -2x_0$. Using this fact and comparing the remaining coefficients gives $a = x_0^2 + 2x_0x_1 = -3x_0^2$ and $b = -x_0^2x_1 = 2x_0^3$. It follows that $4a^3 + 27b^2 = 4(-3x_0^2)^3 + 27(2x_0^3)^2 = 0$.

Note that when $x_0 = x_1$, the polynomial $f(x)$ has a single root of multiplicity 3, and comparing coefficients as above gives $a = b = 4a^3 + 27b^2 = 0$.

Solution to Exercise 2.2

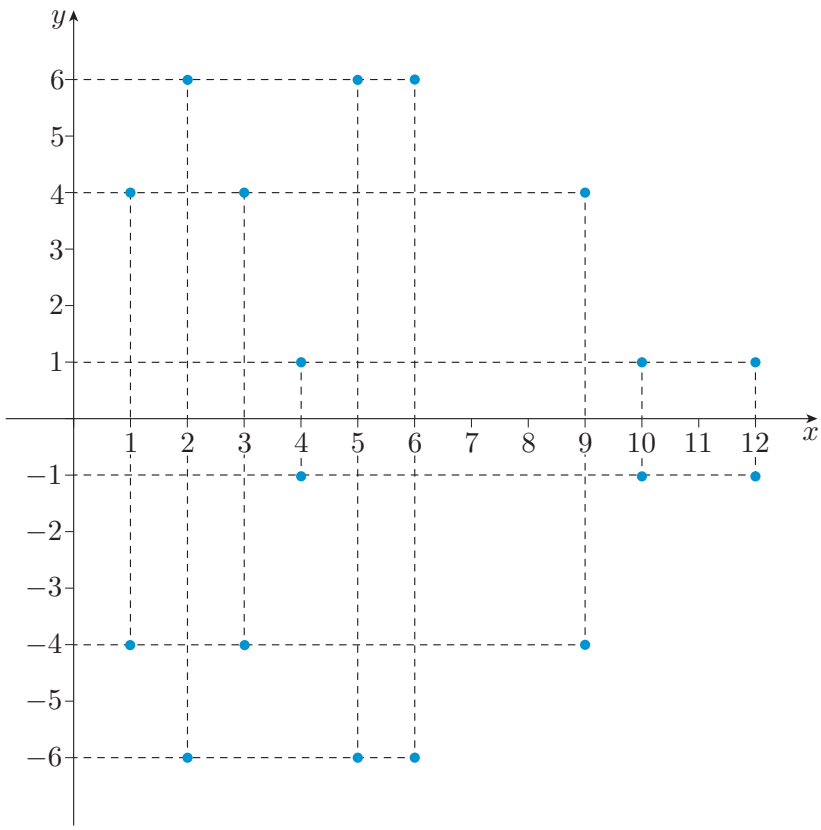
Trying the 11 integer values of x such that $0 \leq x \leq 10$ gives exactly four points on $E_{\mathbb{Q}}(1, 9)$ with integer coordinates $(0, \pm 3)$ and $(8, \pm 23)$.

Solution to Exercise 2.3

We need to identify the set of all $(x, y) \in GF_{13} \times GF_{13}$ such that $y^2 = x^3 + 2$. Since the prime $p = 13$ is not congruent to 2 modulo 3, we cannot use the methods of Example 2.7. However, the field GF_{13} is small enough that for every $x \in GF_{13}$ we can work out $x^3 + 2 \in GF_{13}$.

x	0	1	2	3	4	5	6	7	8	9	10	11	12
$x^3 + 2$	2	3	10	3	1	10	10	7	7	3	1	7	1

By inspection, we find that the set of squares in GF_{13} is $S = \{0, 1, 3, 4, 9, 10, 12\}$. To find the points of $E_{GF_{13}}(0, 2)\backslash\{\mathcal{O}\}$, it suffices to select those $x \in GF_{13}$ such that $x^3 + 2 \in S$, so that the points $(x^3 + 2, \pm\sqrt{x^3 + 2})$ are on the curve. The resulting graph is given below.



A graph of the elliptic curve $E_{GF_{13}}(0,2)\backslash\{\mathcal{O}\}$

Solution to Exercise 2.4

By letting x range over the five possible elements of GF_5, we find that $x^3 + x + 1$ is a square in GF_5 if, and only if, $x \neq 1$. Then

$$E_4 = \{\mathcal{O}, (0, \pm 1), (-1, \pm 2), (2, \pm 1), (-2, \pm 1)\}.$$

We see that $|E_4| = 9$ as in Worked Exercise 2.17, although the underlying fields and the defining polynomials are different. Therefore the candidates for the group $(E_4, +)$ are, again, $\mathbb{Z}_9$ and $\mathbb{Z}_3 \times \mathbb{Z}_3$.

Let us take $P_0 = (0, 1)$ and calculate $2P_0$. Lemma 2.10(b) with $a = b = 1$, $x_0 = 0$ and $y_0 = 1$ gives $c_0 = \frac{1}{2} = -2$, $d_0 = 1$, $x' = 4 = -1$ and $y' = 3 = -2$, so that $2P_0 = -\pi'(P_0) = (-1, 2)$.

Letting $P_1 = P_0 = (0, 1)$ and $P_2 = 2P_0 = (-1, 2)$ and applying Lemma 2.10(a) gives $c = -1$, $d = 1$, $x_3 = 2$ and $y_3 = -1$, so $3P_0 = P_1 + P_2 = -\pi(P_1, P_2) = (2, 1)$.

It follows that the order of the point P_0 in $(E_4, +)$ is different from 3, and therefore the group $(E_4, +)$ must be isomorphic to $\mathbb{Z}_9$. Our calculations also show that P_0 is a generator of this cyclic group.

Solution to Exercise 2.5

We can use the table and the graph of this curve in the solution to Exercise 2.3 or we can follow the method outlined in the solution to Worked Exercise 2.18 to list the points on this curve. We find that

$$E_6 = \{\mathcal{O}\} \cup \{(1, \pm 4), (2, \pm 6), (3, \pm 4), (4, \pm 1), (5, \pm 6), (6, \pm 6), (-4, \pm 4), (-3, \pm 1), (-1, \pm 1)\},$$

and so $|E_6| = 19$. It follows that $(E_6, +) \cong \mathbb{Z}_{19}$. Note that here the underlying field has order 13 only!

Solution to Exercise 2.6

As in the solution to Worked Exercise 2.18, we list the points on the curve. We obtain

$$E_7 = \{\mathcal{O}\} \cup \{(0, \pm 4), (1, \pm 1), (2, \pm 3), (4, \pm 4), (6, 0), (7, \pm 3), (8, \pm 3), (-6, \pm 7), (-4, \pm 4)\},$$

and so $|E_7| = 18$.

By Theorem 2.2 of Book B, Chapter 8, there are just two abelian groups of order 18, namely, $\mathbb{Z}_{18}$ and $\mathbb{Z}_3 \times \mathbb{Z}_6$. To find out which of these is isomorphic to $(E_7, +)$, we experiment with some point, say $P = (0, 4)$. By Lemma 2.10(b), we find that $2P = (4, 4)$ and by part (a) of the same lemma we obtain $3P = (-4, -4)$; a further application of part (b) of the lemma gives $6P = (7, -3)$. Since by Proposition 2.10 of Book B, Chapter 6, the order of every element of $\mathbb{Z}_3 \times \mathbb{Z}_6$ divides 6, it follows that $(E_7, +) \cong \mathbb{Z}_{18}$.

More information is revealed if one uses Lemma 2.10(a) to determine $9P = 6P + 3P = (6, 0)$. The point $(6, 0)$ has order 2 (as the zero y-coordinate implies that the point is equal to its additive inverse) and so $18P = \mathcal{O}$. Since our calculations imply that no smaller multiple of P is equal to $\mathcal{O}$ in this group, we see that P is actually a generator of $(E_7, +) \cong \mathbb{Z}_{18}$.

Solution to Exercise 2.7

We follow the method outlined in Example 2.21. With the help of parts (a) and (b) of Lemma 2.10, Alice calculates

$$M_1 = kP = 5P = (1, -1) \quad \text{and}$$
$$M_2 = M + kP_B = (8, -3) + 5(-4, -4) = (0, 4),$$

and sends both ciphertexts M_1 and M_2 to Bob, who then calculates

$$M_2 - s_B M_1 = (0, 4) - 3(1, -1) = (8, -3),$$

which is Alice's original message.

With the extra knowledge of the solution to Exercise 2.6, however, one can avoid most of the applications of Lemma 2.10 and instead use the fact that the point $P = (0, 4)$ is a generator for the group $(E, +)$.

Solution to Exercise 2.8

We follow the method outlined in Example 2.22. Using parts (a) and (b) of Lemma 2.10, Alice computes $kP = 4(0, 4) = (8, -3)$ and $kP_B = 4(-4, -4) = (7, 3) = (d_1, d_2)$. She then determines $c_1 = d_1 m_1 = 14 \equiv -3 \pmod{17}$ and $c_2 = d_2 m_2 = 12 \equiv -5 \pmod{17}$. Therefore Alice's ciphertext is $(kP, c_1, c_2) = ((8, -3), -3, -5)$.

After receiving the ciphertext, Bob calculates $s_B(kP) = 3(8, -3) = (7, 3)$ and obtains $d_1 = 7$ and $d_2 = 3$. He then determines $d_1^{-1} = 5$ and $d_2^{-1} = 6$ in GF_{17}. Finally, he computes

$$c_1 d_1^{-1} = (-3) \times 5 \equiv 2 \pmod{17} \quad \text{and}$$
$$c_2 d_2^{-1} = (-5) \times 6 \equiv 4 \pmod{17},$$

thus recovering Alice's intended message $(2, 4)$.

We again point out that the extra knowledge of the solution to Exercise 2.6 allows one to avoid most of the applications of Lemma 2.10 by using the fact that the point $P = (0, 4)$ is a generator for the group $(E, +)$.